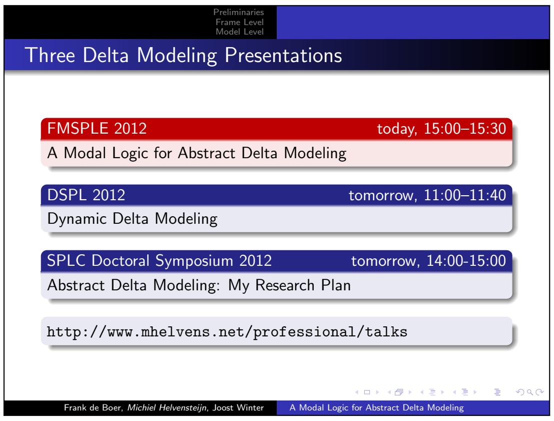
A Modal Logic for Abstract Delta Modeling

Frank de Boer *Michiel Helvensteijn* Joost Winter

FMSPLE, 02-09-2012

Introduction

This is the presentation of the paper *A Modal Logic for Abstract Delta Modeling* as presented at FMSPLE 2012. The paper itself has more information and can be found in the SPLC 2012 workshop proceedings.



Preliminaries
Frame Level
Model Level

Three Delta Modeling Presentations

FMSPLE 2012 today, 15:00–15:30
A Modal Logic for Abstract Delta Modeling

DSPL 2012 tomorrow, 11:00–11:40
Dynamic Delta Modeling

SPLC Doctoral Symposium 2012 tomorrow, 14:00-15:00
Abstract Delta Modeling: My Research Plan

`http://www.mhelvens.net/professional/talks`

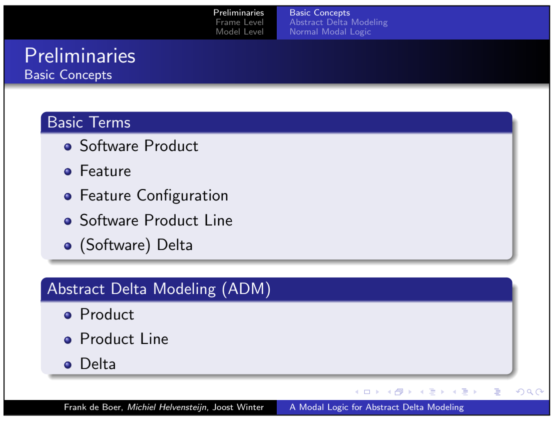
Frank de Boer, *Michiel Helvensteijn*, Joost Winter — A Modal Logic for Abstract Delta Modeling

As of writing this, I am going to present three papers at SPLC colocated workshops. (If you are reading this after September 3rd, I assume I will already have done so.)

For slides and handouts of this, and the other two presentations, see my website: `http://www.mhelvens.net/professional/talks`

1 Preliminaries

1.1 Basic Concepts



Preliminaries
Frame Level
Model Level

Basic Concepts
Abstract Delta Modeling
Normal Modal Logic

Preliminaries
Basic Concepts

Basic Terms
- Software Product
- Feature
- Feature Configuration
- Software Product Line
- (Software) Delta

Abstract Delta Modeling (ADM)
- Product
- Product Line
- Delta

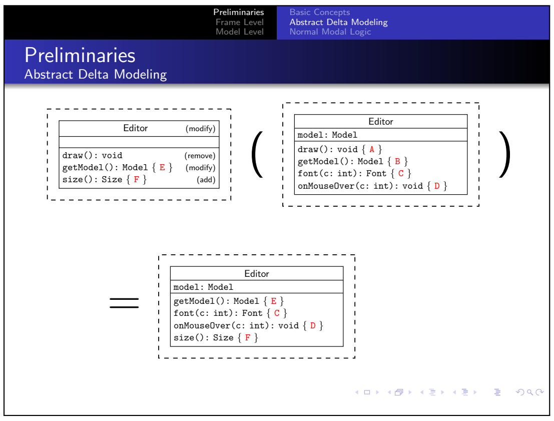
Frank de Boer, *Michiel Helvensteijn*, Joost Winter — A Modal Logic for Abstract Delta Modeling

A *software product* is basically just a software system or program. A *feature* is some characteristic we may want a software product to have. A *feature configuration* is a legal combination of such features. A *software product line* is a set of related software products, one for each feature configuration. A *software delta* (somtimes just called 'delta', but in this talk, we need the distinction) can transform one software product into another software product, and can thus be used to organize the code in a software product line so that any product from the product line can be mechanically generated given the desired feature configuration. This is called *Delta Modeling.*

Abstract Delta Modeling is an abstract algebraic description of delta modeling (which also introduces some novel techniques). It abstracts away from any concept of programming code and presents more general notions of *product* and *product line.* Delta Modeling works by selectively applying deltas to a core product, generating any specific product of the product line.

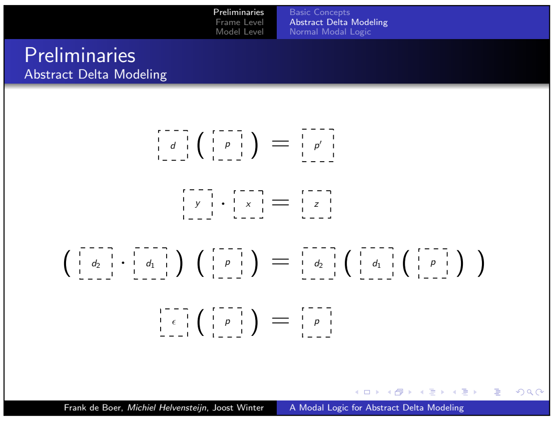
Feature configurations don't play a big role in this paper or this presentation, but they're here for completeness sake. Features labels, for us, will play the role of *semantic judgements* about or specifications of products, not the role of *selecting* a specific product.

1.2 Abstract Delta Modeling



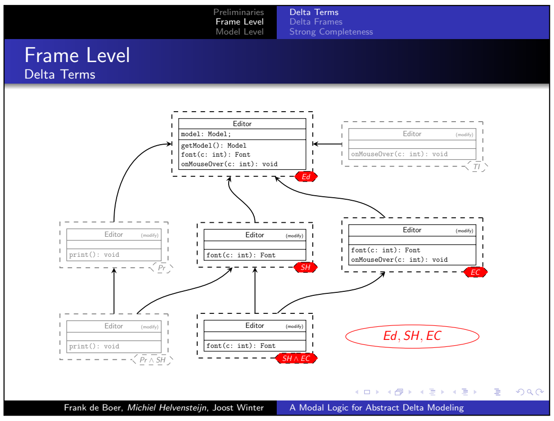
The basic building blocks of (abstract) delta modeling are *products* and *deltas* that transform one product into another product.

For instance, here you see a product (top right) containing a single class being transformed on a method level by a delta (top left). We call this *delta application*.



Object oriented programming is the domain delta modeling was primarily designed for, and it is a nice domain for examples, since practically all computer scientists are familiar with OOP. But ADM is designed on a purely abstract level and doesn't really say anything about software at all.

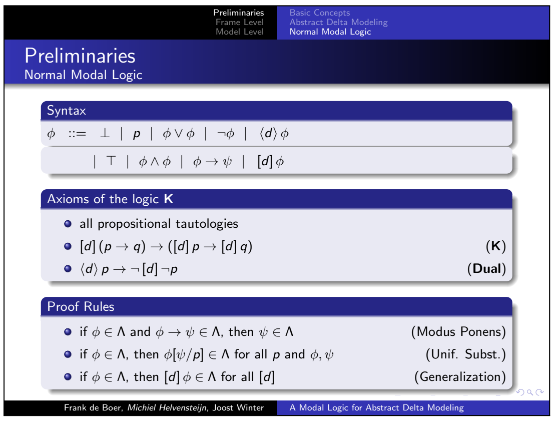
The top formula on this slide shows the abstract version of the previous slide. The second line shows the *sequential composition* operation. The third and fourth line show the two axioms that makes delta application a *monoid action*. ϵ represents the 'neutral' or 'empty' delta.



A product line developed using delta modeling has a set of deltas which should be selectively applied to a core product. Each delta is annotated with an *application condition*, which determines for which feature configurations it should be applied. They are also put in a partial order, which restricts their order of aplication.

For example, in this slide you see that the feature configuration $\{Ed, SH, EC\}$ selects four deltas, which must be applied in one of two valid linear orders.

1.3 Normal Modal Logic



Preliminaries | Basic Concepts
Frame Level | Abstract Delta Modeling
Model Level | Normal Modal Logic

Preliminaries
Normal Modal Logic

Syntax

$\phi \;::=\; \bot \mid p \mid \phi \vee \phi \mid \neg\phi \mid \langle d\rangle\,\phi$

$\quad\mid \top \mid \phi\wedge\phi \mid \phi\to\psi \mid [d]\,\phi$

Axioms of the logic K

- all propositional tautologies
- $[d]\,(p\to q)\to([d]\,p\to[d]\,q)$ **(K)**
- $\langle d\rangle\,p\to\neg\,[d]\,\neg p$ **(Dual)**

Proof Rules

- if $\phi\in\Lambda$ and $\phi\to\psi\in\Lambda$, then $\psi\in\Lambda$ (Modus Ponens)
- if $\phi\in\Lambda$, then $\phi[\psi/p]\in\Lambda$ for all p and ϕ,ψ (Unif. Subst.)
- if $\phi\in\Lambda$, then $[d]\,\phi\in\Lambda$ for all $[d]$ (Generalization)

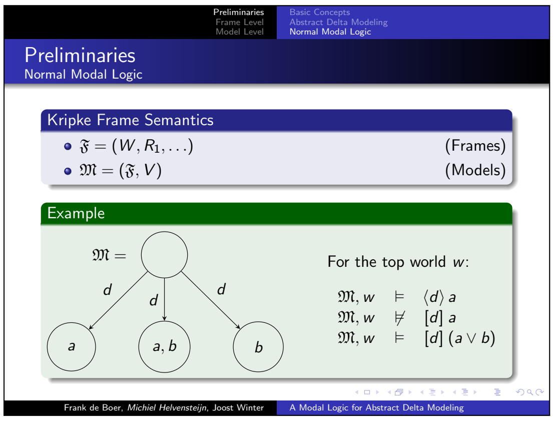
Frank de Boer, *Michiel Helvensteijn*, Joost Winter | A Modal Logic for Abstract Delta Modeling

Modal logic is useful for reasoning about relations between '*worlds*', binary relations in particular. Well known examples are temporal logic, which reasons about past and future, epistemic logic, which reasons about knowledge of agents and dynamic logic, which reasons about the effect of programs on a state.

The syntax of the basic normal modal logic **K** is basically propositional logic, with the added notation of a unary modality operator $\langle d\rangle$, labeled with binary relation d. The second set of syntactic constructs can be defined in terms of the first set. In particular, see the axiom **Dual**.

Intuitively, $\langle d\rangle\,\phi$ means that from 'current world', at least one world can be reached with d in which ϕ holds, but ϕ doesn't necessarily hold in all worlds reachable with d. $\langle d\rangle$ feels like $\exists$.

$[d]\,\phi$ means that in every world reachable with d, we have ϕ, but it is possible that d reaches no worlds at all. $[D]$ feels like $\forall$.



Preliminaries
Frame Level
Model Level

Basic Concepts
Abstract Delta Modeling
Normal Modal Logic

Preliminaries
Normal Modal Logic

Kripke Frame Semantics

- $\mathfrak{F} = (W, R_1, \ldots)$ (Frames)
- $\mathfrak{M} = (\mathfrak{F}, V)$ (Models)

Example

For the top world w:

$\mathfrak{M}, w \models \langle d \rangle\, a$
$\mathfrak{M}, w \not\models [d]\, a$
$\mathfrak{M}, w \models [d]\, (a \vee b)$

$\mathfrak{M} =$
d
d
d
a
a, b
b

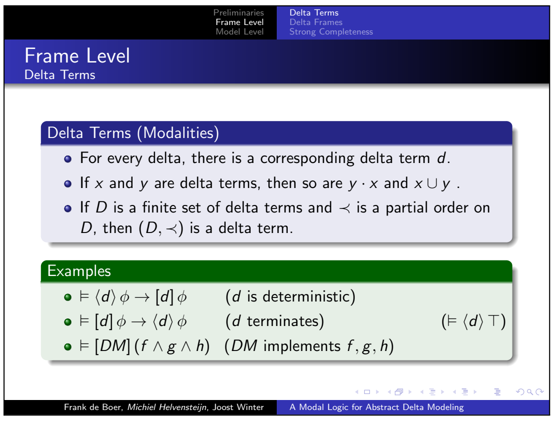
Frank de Boer, *Michiel Helvensteijn*, Joost Winter — A Modal Logic for Abstract Delta Modeling

Modal logics are given semantics through *Kripke Frames*. A *frame* $\mathfrak{F}$ is a set of worlds, and a set of relations between those worlds. A *model* $\mathfrak{M}$ is a frame in which all worlds have a set of proposition letters assigned to them.

The diagram gives an example of such a model $\mathfrak{M}$. The nodes are worlds and the arrows represent the binary relations. The nodes contain the proposition letters that are true in that world. To give you an intuition of the logic, the slide shows three example judgments. Only the first and the third hold. The second doesn't hold, because there is a world, reachable through d, in which a does not hold.

2 Frame Level

2.1 Delta Terms



Preliminaries
Frame Level
Model Level

Delta Terms
Delta Frames
Strong Completeness

Frame Level
Delta Terms

Delta Terms (Modalities)

- For every delta, there is a corresponding delta term d.
- If x and y are delta terms, then so are $y \cdot x$ and $x \cup y$.
- If D is a finite set of delta terms and $\prec$ is a partial order on D, then $(D, \prec)$ is a delta term.

Examples

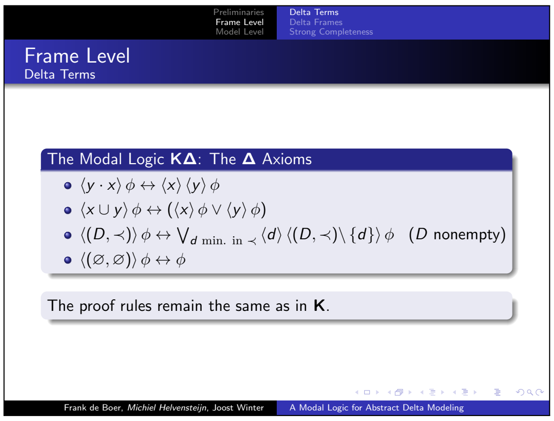
- $\models \langle d \rangle\, \phi \rightarrow [d]\, \phi$ (d is deterministic)
- $\models [d]\, \phi \rightarrow \langle d \rangle\, \phi$ (d terminates) ($\models \langle d \rangle\, \top$)
- $\models [DM]\, (f \wedge g \wedge h)$ (DM implements f, g, h)

Frank de Boer, *Michiel Helvensteijn*, Joost Winter — A Modal Logic for Abstract Delta Modeling

In our modal logic, we see products as worlds and deltas as binary relations between those worlds. Deltas are usually deterministic (functional) and terminating entities, but in this formalism, they don't have to be.

Delta terms are the syntactic counterpart of real deltas, used in our logic in order to strictly separate syntax and semantics. There are three natural ways to combine delta terms in order to form more complex modalities. *Composition* and *union* of relations should be clear enough. The third compound delta term represents a partially ordered set of deltas. Semantically, it applies all deltas in D in any linear extension of $\prec$. Both union and partial ordering can potentially introduce nondeterminism.

The slide shows three examples of properties that can be expressed about deltas using this syntax.



Preliminaries
Frame Level
Model Level

Delta Terms
Delta Frames
Strong Completeness

Frame Level
Delta Terms

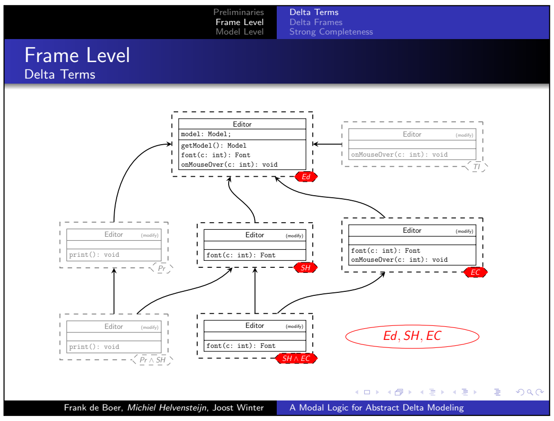
The Modal Logic $\mathbf{K\Delta}$: The $\mathbf{\Delta}$ Axioms

- $\langle y \cdot x \rangle\, \phi \leftrightarrow \langle x \rangle\, \langle y \rangle\, \phi$
- $\langle x \cup y \rangle\, \phi \leftrightarrow (\langle x \rangle\, \phi \vee \langle y \rangle\, \phi)$
- $\langle (D, \prec) \rangle\, \phi \leftrightarrow \bigvee_{d \text{ min. in } \prec} \langle d \rangle\, \langle (D, \prec) \setminus \{d\} \rangle\, \phi$ (D nonempty)
- $\langle (\varnothing, \varnothing) \rangle\, \phi \leftrightarrow \phi$

The proof rules remain the same as in $\mathbf{K}$.

Frank de Boer, *Michiel Helvensteijn*, Joost Winter — A Modal Logic for Abstract Delta Modeling

We add now a minimal set of axioms which allow you to compose or decompose modalities. With the **K** axioms, the $\mathbf{K\Delta}$ axioms and the original proofrules of **K**, we have the new frame level proof system of our new logic.



Preliminaries
Frame Level
Model Level

Delta Terms
Delta Frames
Strong Completeness

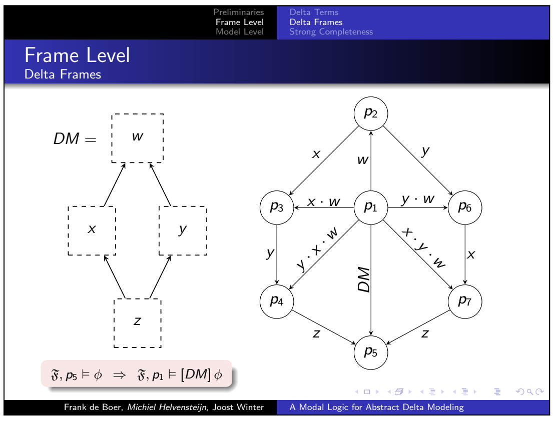
Frame Level
Delta Terms

Frank de Boer, *Michiel Helvensteijn*, Joost Winter — A Modal Logic for Abstract Delta Modeling

So we should now be able to formally ask a question like: Are the features *Ed*, *SH* and *EC* actually implemented by the product we get if we apply these selected deltas? They should be, of course. Otherwise those deltas are just not well designed. A more general question: Is this true for every valid feature configuration? This property, called *product line completeness*, is an important one to verify.

Of course, we need some more concrete feature specifications to determine such a thing in practice. But there's a fair amount we can do on an abstract level.

2.2 Delta Frames



Preliminaries
Frame Level
Model Level

Delta Terms
Delta Frames
Strong Completeness

Frame Level
Delta Frames

$$DM = \begin{tikzcd} & \boxed{w} & \\ \boxed{x} \arrow[ur] & & \boxed{y} \arrow[ul] \\ & \boxed{z} \arrow[ul] \arrow[ur] & \end{tikzcd}$$

$$\begin{tikzcd} & p_2 \arrow[dl, "x"'] \arrow[dr, "y"] & \\ p_3 \arrow[d, "y"'] & p_1 \arrow[u, "w"] \arrow[l, "x \cdot w"'] \arrow[r, "y \cdot w"] \arrow[dl, "y \cdot x \cdot w"] \arrow[dr, "x \cdot y \cdot w"] \arrow[dd, "DM"] & p_6 \arrow[d, "x"] \\ p_4 \arrow[dr, "z"'] & & p_7 \arrow[dl, "z"] \\ & p_5 & \end{tikzcd}$$

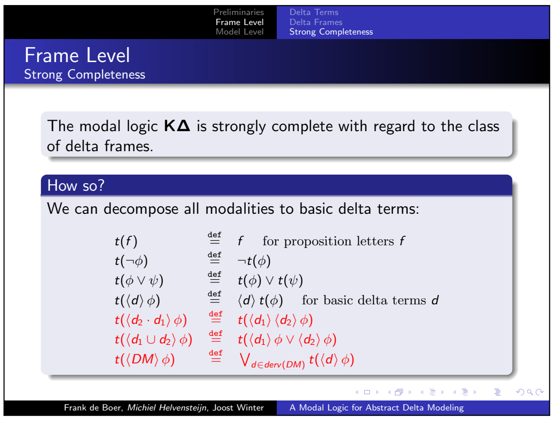
$\mathfrak{F}, p_5 \vDash \phi \quad \Rightarrow \quad \mathfrak{F}, p_1 \vDash [DM]\,\phi$

Frank de Boer, *Michiel Helvensteijn*, Joost Winter | A Modal Logic for Abstract Delta Modeling

The semantics of our new logic is given by Kripke frames again, but this time with the new compound delta relations in the mix. On the right of this slide, we have a partial *delta frame*, showing the parts relevant for applying the given delta model DM (on the left) to 'core product' p_1. As expected, the nodes of the frame represent products and the arrows represent deltas.

The shown relations show basic delta terms, as well as composed terms and a partially ordered term (DM). DM could also be written as $z \cdot y \cdot x \cdot w \cup z \cdot x \cdot y \cdot w$, completing our set with union. Note that not nearly all relations are shown in this diagram. We didn't show $y \cdot x \ni (p_2, p_4)$, nor what could happen if we applied, e.g., delta w to product p_4. Depending on the concrete nature of the given products and deltas, this frame may not be finite.

2.3 Strong Completeness



Preliminaries
Frame Level
Model Level

Delta Terms
Delta Frames
Strong Completeness

Frame Level
Strong Completeness

The modal logic **KΔ** is strongly complete with regard to the class of delta frames.

How so?

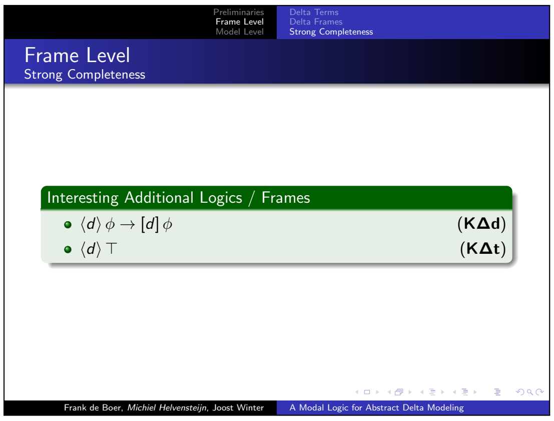
We can decompose all modalities to basic delta terms:

$$\begin{aligned} t(f) &\overset{\text{def}}{=} f \quad \text{for proposition letters } f \\ t(\neg\phi) &\overset{\text{def}}{=} \neg t(\phi) \\ t(\phi \vee \psi) &\overset{\text{def}}{=} t(\phi) \vee t(\psi) \\ t(\langle d \rangle\, \phi) &\overset{\text{def}}{=} \langle d \rangle\, t(\phi) \quad \text{for basic delta terms } d \\ t(\langle d_2 \cdot d_1 \rangle\, \phi) &\overset{\text{def}}{=} t(\langle d_1 \rangle\, \langle d_2 \rangle\, \phi) \\ t(\langle d_1 \cup d_2 \rangle\, \phi) &\overset{\text{def}}{=} t(\langle d_1 \rangle\, \phi \vee \langle d_2 \rangle\, \phi) \\ t(\langle DM \rangle\, \phi) &\overset{\text{def}}{=} \bigvee_{d \in derv(DM)} t(\langle d \rangle\, \phi) \end{aligned}$$

Frank de Boer, *Michiel Helvensteijn*, Joost Winter | A Modal Logic for Abstract Delta Modeling

We proved that our logic is strongly complete with regard to the class of all delta frames. So everything that's true on a frame level can be proved within our logic.

This is due to the simplicity of our modalities. Our partially ordered sets of deltas are always finite and we do not have a concept of iteration (such as in dynamic logic). So we can translate any formula in **KΔ** to an equivalent one with no compound delta terms. Basically, it becomes a **K** formula and we prove completeness by invoking completeness of **K** with regard to the class of all frames.



Preliminaries
Frame Level
Model Level

Delta Terms
Delta Frames
Strong Completeness

Frame Level
Strong Completeness

Interesting Additional Logics / Frames

- $\langle d \rangle\, \phi \to [d]\, \phi$ **(KΔd)**
- $\langle d \rangle\, \top$ **(KΔt)**

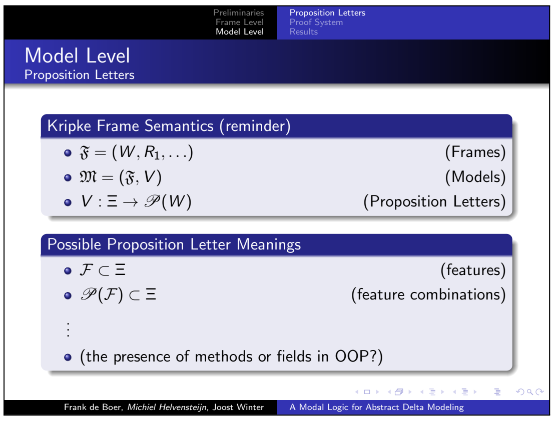
Frank de Boer, *Michiel Helvensteijn*, Joost Winter | A Modal Logic for Abstract Delta Modeling

Here are some interesting additional frames and corresponding strongly complete logics. The logics may be generated by adding the given formula to the set of axiom schemata.

(Yes, I stole these formulas from a previous slide.)

3 Model Level

3.1 Proposition Letters



Preliminaries
Frame Level
Model Level

Proposition Letters
Proof System
Results

Model Level
Proposition Letters

Kripke Frame Semantics (reminder)

- $\mathfrak{F} = (W, R_1, \dots)$ (Frames)
- $\mathfrak{M} = (\mathfrak{F}, V)$ (Models)
- $V : \Xi \to \mathscr{P}(W)$ (Proposition Letters)

Possible Proposition Letter Meanings

- $\mathcal{F} \subset \Xi$ (features)
- $\mathscr{P}(\mathcal{F}) \subset \Xi$ (feature combinations)

$\vdots$

- (the presence of methods or fields in OOP?)

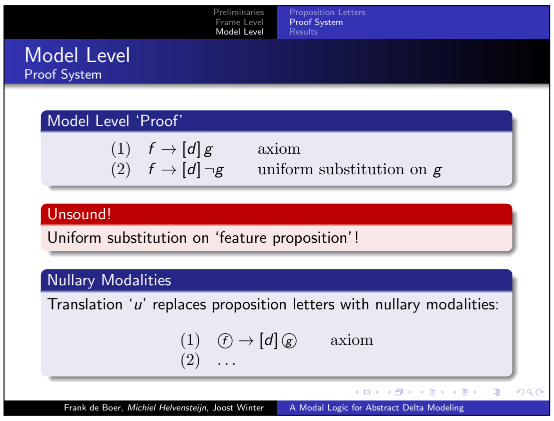
Frank de Boer, *Michiel Helvensteijn*, Joost Winter | A Modal Logic for Abstract Delta Modeling

Now we need to determine what our *proposition letters* are going to represent, to give us something useful to reason about on a model level. What do we want to specify or prove about the effect of a delta on a product?

There are many possibilities. We've already mentioned features. That is, of course, particularly relevant to SPL research. It may not be enough, if you want to reason about feature interaction. So we could reason about feature combinations instead, as seen in the second item of this slide. You would want an extra axiom: $F \cup G \to F \wedge G$. This is the approach taken in the Delta Modeling Workflow paper.

Those are still quite high-level, though. Maybe you also want to reason about lower level concepts, such as the presence of specific methods and fields inside a class. This way, every concrete domain could define its own 'sub-logic'.

3.2 Proof System



Preliminaries
Frame Level
Model Level

Proposition Letters
Proof System
Results

Model Level
Proof System

Model Level 'Proof'

(1)	$f \to [d]\, g$	axiom
(2)	$f \to [d]\, \neg g$	uniform substitution on g

Unsound!

Uniform substitution on 'feature proposition'!

Nullary Modalities

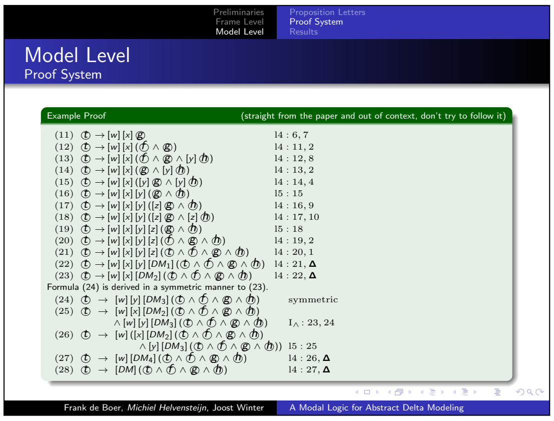
Translation 'u' replaces proposition letters with nullary modalities:

(1)	$\textcircled{f} \to [d]\, \textcircled{g}$	axiom
(2)	$\dots$	

Frank de Boer, *Michiel Helvensteijn*, Joost Winter | A Modal Logic for Abstract Delta Modeling

Simply using the proof system from the frame level doesn't work. The 'proof' on this slide is obviously faulty.

Now that we've assigned meaning to proposition letters and allow them into our axioms, we are not allowed to arbitrarily use uniform substitution on them anymore. For the purpose of proofs, we replace our 'meaningful' proposition letters (not the ones introduced by propositional tautologies) with corresponding *nullary modalities*, which may be seen as propositional constants. That way, we **can** use the frame level proof system again.



Preliminaries
Frame Level
Model Level

Proposition Letters
Proof System
Results

Model Level
Proof System

Example Proof (straight from the paper and out of context, don't try to follow it)

(11)	$\textcircled{t} \to [w]\,[x]\,\textcircled{g}$	l4 : 6, 7
(12)	$\textcircled{t} \to [w]\,[x]\,(\textcircled{f} \wedge \textcircled{g})$	l4 : 11, 2
(13)	$\textcircled{t} \to [w]\,[x]\,(\textcircled{f} \wedge \textcircled{g} \wedge [y]\,\textcircled{h})$	l4 : 12, 8
(14)	$\textcircled{t} \to [w]\,[x]\,(\textcircled{g} \wedge [y]\,\textcircled{h})$	l4 : 13, 2
(15)	$\textcircled{t} \to [w]\,[x]\,([y]\,\textcircled{g} \wedge [y]\,\textcircled{h})$	l4 : 14, 4
(16)	$\textcircled{t} \to [w]\,[x]\,[y]\,(\textcircled{g} \wedge \textcircled{h})$	l5 : 15
(17)	$\textcircled{t} \to [w]\,[x]\,[y]\,([z]\,\textcircled{g} \wedge \textcircled{h})$	l4 : 16, 9
(18)	$\textcircled{t} \to [w]\,[x]\,[y]\,([z]\,\textcircled{g} \wedge [z]\,\textcircled{h})$	l4 : 17, 10
(19)	$\textcircled{t} \to [w]\,[x]\,[y]\,[z]\,(\textcircled{g} \wedge \textcircled{h})$	l5 : 18
(20)	$\textcircled{t} \to [w]\,[x]\,[y]\,[z]\,(\textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h})$	l4 : 19, 2
(21)	$\textcircled{t} \to [w]\,[x]\,[y]\,[z]\,(\textcircled{t} \wedge \textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h})$	l4 : 20, 1
(22)	$\textcircled{t} \to [w]\,[x]\,[y]\,[DM_1]\,(\textcircled{t} \wedge \textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h})$	l4 : 21, $\boldsymbol{\Delta}$
(23)	$\textcircled{t} \to [w]\,[x]\,[DM_2]\,(\textcircled{t} \wedge \textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h})$	l4 : 22, $\boldsymbol{\Delta}$

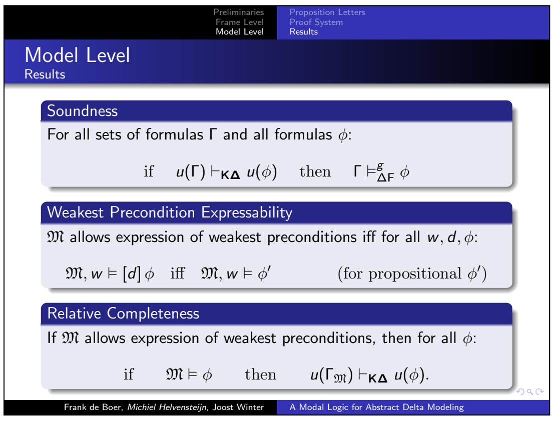
Formula (24) is derived in a symmetric manner to (23).

(24)	$\textcircled{t} \to [w]\,[y]\,[DM_3]\,(\textcircled{t} \wedge \textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h})$	symmetric
(25)	$\textcircled{t} \to [w]\,[x]\,[DM_2]\,(\textcircled{t} \wedge \textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h})$ $\wedge [w]\,[y]\,[DM_3]\,(\textcircled{t} \wedge \textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h})$	$\mathrm{I}_\wedge$: 23, 24
(26)	$\textcircled{t} \to [w]\,([x]\,[DM_2]\,(\textcircled{t} \wedge \textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h})$ $\wedge [y]\,[DM_3]\,(\textcircled{t} \wedge \textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h}))$	l5 : 25
(27)	$\textcircled{t} \to [w]\,[DM_4]\,(\textcircled{t} \wedge \textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h})$	l4 : 26, $\boldsymbol{\Delta}$
(28)	$\textcircled{t} \to [DM]\,(\textcircled{t} \wedge \textcircled{f} \wedge \textcircled{g} \wedge \textcircled{h})$	l4 : 27, $\boldsymbol{\Delta}$

Frank de Boer, *Michiel Helvensteijn*, Joost Winter | A Modal Logic for Abstract Delta Modeling

This slide shows an example proof on the model level using this technique. For more details about this particular proof, I advice you to take a look at the paper.

3.3 Results


Preliminaries
Frame Level
Model Level

Proposition Letters
Proof System
Results

Model Level
Results

Soundness
For all sets of formulas Γ and all formulas ϕ:

$$\text{if} \quad u(\Gamma) \vdash_{\mathbf{K\Delta}} u(\phi) \quad \text{then} \quad \Gamma \vDash^{g}_{\Delta\mathsf{F}} \phi$$

Weakest Precondition Expressability
$\mathfrak{M}$ allows expression of weakest preconditions iff for all w, d, ϕ:

$$\mathfrak{M}, w \vDash [d]\,\phi \quad \text{iff} \quad \mathfrak{M}, w \vDash \phi' \qquad \text{(for propositional } \phi'\text{)}$$

Relative Completeness
If $\mathfrak{M}$ allows expression of weakest preconditions, then for all ϕ:

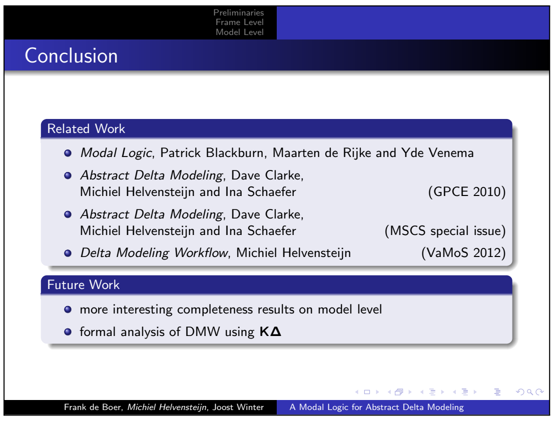
$$\text{if} \quad \mathfrak{M} \vDash \phi \quad \text{then} \quad u(\Gamma_{\mathfrak{M}}) \vdash_{\mathbf{K\Delta}} u(\phi).$$

Frank de Boer, *Michiel Helvensteijn*, Joost Winter | A Modal Logic for Abstract Delta Modeling

We proved that this proof system is sound on a model level. More details may be found in the paper.

We also have a form of completeness on a specific subset of models. Those are the models that allow expression of weakest precondition in a purely propositional way. (Preconditions can of course always be expressed using $\phi = [d]\,\psi$.) If this is always possible, we can transform any formula into an equivalent formula in propositional logic, and so our completeness result is proved using the completeness of propositional logic.

Conclusion


Preliminaries
Frame Level
Model Level

Conclusion

Related Work
- *Modal Logic*, Patrick Blackburn, Maarten de Rijke and Yde Venema
- *Abstract Delta Modeling*, Dave Clarke, Michiel Helvensteijn and Ina Schaefer (GPCE 2010)
- *Abstract Delta Modeling*, Dave Clarke, Michiel Helvensteijn and Ina Schaefer (MSCS special issue)
- *Delta Modeling Workflow*, Michiel Helvensteijn (VaMoS 2012)

Future Work
- more interesting completeness results on model level
- formal analysis of DMW using $\mathbf{K\Delta}$

Frank de Boer, *Michiel Helvensteijn*, Joost Winter | A Modal Logic for Abstract Delta Modeling

Modal logic is described very nicely in the book ‘Modal Logic’ by Blackburn, de Rijke and Venema. Our notation is based on the notation from this book. Abstract Delta Modeling (ADM) is described in great detail in the two papers bearing that title. The theory in this paper is fully compatible with ADM, except that deltas do not have to be deterministic and terminating here. The logic we have defined here will be very useful in proving properties about the Delta Modeling Workflow more succinctly.

Possible future work includes more general completeness results on the model level and the use of the modal logic for analyzing the Delta Modeling Workflow.

Preliminaries
Frame Level
Model Level

Conclusion

`www.mhelvens.net`

Frank de Boer, *Michiel Helvensteijn*, Joost Winter | A Modal Logic for Abstract Delta Modeling

Thanks for reading!