A Modal Logic for Abstract Delta Modeling

Frank de Boer*
CWI, Amsterdam
Leiden University
The Netherlands
f.s.de.boer@cwi.nl

Michiel Helvensteijn*
CWI, Amsterdam
Leiden University
The Netherlands
michiel.helvensteijn@cwi.nl

Joost Winter
CWI, Amsterdam
The Netherlands
j.winter@cwi.nl

ABSTRACT

Abstract Delta Modeling is a technique for implementing (software) product lines. Deltas are placed in a partial order which restricts their application and are then sequentially applied to a core product in order to form specific products in the product line. In this paper we explore the semantics of deltas in more detail. We regard them as relations between products and introduce a multimodal logic that may be used for reasoning about their effects. Our main innovation is a modality for partially ordered sets of deltas. We prove completeness results on both the frame level and the model level and demonstrate the logic through an example.

Categories and Subject Descriptors

D.2.11 [**Software Engineering**]: Software Architectures; F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*Modal Logic*

General Terms

Theory, Languages

Keywords

Modal logic, delta modeling, software product lines

1. INTRODUCTION

Delta Modeling [14, 15, 13] is a technique for implementing *software product lines* [12]: a way to optimally reuse code between software products which differ only by which features they support. The code is divided into units called *deltas*, which can incrementally transform a core product in order to generate a product in the product line.

Clarke et al. [5, 6] described delta modeling in an abstract algebraic manner called the *Abstract Delta Modeling* (ADM) approach. In that work, delta modeling is not restricted to *software* product lines, but rather product lines of *any* domain. It gives a formal description of deltas, how

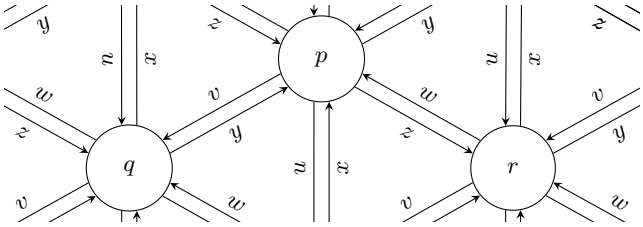

Figure 1: Example view of a delta frame with products p, q, r and deltas u, v, w, x, y, z currently visible

they can be applied to *products*, how they can be combined, how they can be linked to features from the feature model, as well as how to avoid or resolve implementation conflicts. Most notably, a partial order is imposed on the deltas restricting their order of application. This allows for an exact specification of dependency between deltas, as well as the implementation of desired feature interaction and the resolution of conflicts with a minimum of code duplication.

At its core, ADM is about deltas that can transform one product into another product. We need a way to specify and reason about the semantics of deltas, and what effect they have on the features that are supported by a product. We want to be able to specify that a delta implements a specific feature or that a delta refrains from breaking an existing feature. We want to prove that if certain local guarantees are met, that specific global properties, such as *product line completeness* [9, 10], are then guaranteed to hold.

In this paper we introduce a modal logic in order to reason about the semantics of deltas. Basically, we take the set of all possible products as the set of *worlds* in our frame (Figure 1). We then model deltas as binary relations on this set. In previous work, all deltas were deterministic (functional). We now generalize the notion of a delta, and allow them to be non-deterministic as well as non-terminating. In our logic, we want to be able to make judgments such as

$$\vDash \langle d \rangle\, f \qquad\qquad \vDash [d]\, f$$

meaning "delta d may implement feature f" (left) and "delta d must implement feature f" (right). Or perhaps, for all ϕ:

$$\vDash \langle d \rangle\, \phi \to [d]\, \phi \qquad\qquad \vDash [d]\, \phi \to \langle d \rangle\, \phi$$

meaning "delta d is deterministic" (left) and "delta d always terminates" (right). Note that we implicitly quantify over all products that the delta may be applied to.

We also introduce additional modalities, representing *delta models* (partially ordered sets of deltas, Definition 4), in order to make judgments such as

$$\vDash [DM]\, (f \wedge g \wedge h)$$

meaning "delta model DM implements features f, g and h".

*These authors are partially supported by the EU project FP7-231620 HATS: Highly Adaptable and Trustworthy Software using Formal Models (http://www.hats-project.eu)

SPLC - Vol. II September 02 - 07 2012, Salvador, Brazil

The paper is structured as follows. Sections 2 and 3 summarize the relevant theory of abstract delta modeling and modal logic respectively. Section 4 introduces the syntax and semantics of our modal logic and proves strong completeness with regard to our frames. Section 5 introduces proposition letters and explores our logic on a model level. Section 6 concludes and discusses related and future work.

2. ABSTRACT DELTA MODELING

To make this paper self-contained, we now repeat the relevant theory from ADM. For more detailed information, we refer the reader to [5, 6].

2.1 Products and Deltas

First, we assume a set of *products*, $\mathcal{P}$. The set of possible modifications to products forms a delta monoid, as follows:

Definition 1 (Delta Monoid). *A* delta monoid *is a monoid* $(\mathcal{D}, \cdot, \epsilon)$*, where* $\mathcal{D}$ *is a set of product modifications (referred to as* deltas*), and the operation* $\cdot : \mathcal{D} \times \mathcal{D} \to \mathcal{D}$ *corresponds to their sequential composition.* $y \cdot x$ *denotes the modification applying first* x *and then* y*. The neutral element* ϵ *of the monoid corresponds to modifying nothing.*

Applying a delta to a product results in another product. This is captured by the notion of delta action. The following definition differs from previous work [5, 6], in which deltas were always deterministic, and would always terminate. The notion of nondeterministic delta action allows for both nondeterminism and non-termination, by resulting in a set of products, rather than a single product.

Definition 2 (Nondeterministic Delta Action). *A* nondeterministic delta action *is an operation* $-(-) : \mathcal{D} \times \mathcal{P} \to \mathscr{P}(\mathcal{P})$*. If* $x \in \mathcal{D}$ *and* $p \in \mathcal{P}$*, then* $x(p) \subseteq \mathcal{P}$ *is the set of products that may result from applying delta* x *to product* p*. It satisfies the conditions* $(y \cdot x)(p) = \bigcup_{q \in x(p)} y(q)$ *and* $\epsilon(p) = \{p\}$*.*

This all leads to the notion of a *deltoid*, which describes all building blocks necessary to create a product line in a concrete domain.

Definition 3 (Deltoid). *A* deltoid *is a quintuple* $(\mathcal{P}, \mathcal{D}, \cdot, \epsilon, -(-))$*, where* $\mathcal{P}$ *is a product set,* $(\mathcal{D}, \cdot, \epsilon)$ *is a delta monoid and* $-(-)$ *is a nondeterministic delta action.*

2.2 Delta Models

A delta model describes the set of deltas required to build a specific product, along with a strict partial order on those deltas, restricting the order in which they may be applied.

Definition 4 (Delta Model). *A* delta model *is a pair* $(D, \prec)$*, where* $D \subseteq \mathcal{D}$ *is a finite set of deltas and* $\prec \subseteq D \times D$ *is a strict partial order on* D*.* $x \prec y$ *states that* x *must be applied before* y*, though not necessarily directly before.*

The partial order represents the intuition that a delta applied later has full access to earlier deltas and more authority over modifications to the product.

The semantics of a delta model is defined by its derivations. A *derivation* is a delta formed by a sequential composition of all deltas from D, in some linearization of the partial order.

Definition 5 (Derivation). *Given a delta model* $DM = (D, \prec)$*, its* derivations *are defined to be*

$$derv(DM) \stackrel{\text{def}}{=} \left\{ x_n \cdot \ldots \cdot x_1 \;\middle|\; \begin{array}{l} x_1, \ldots, x_n \text{ is a linear} \\ \text{extension of } (D, \prec) \end{array} \right\}.$$

Observe that when D is empty, $derv(DM) = \{\epsilon\}$. Also note that $derv(DM)$ may potentially generate more than one distinct derivation, as non-commutative deltas may be applied in different orders. Techniques for ensuring a unique derivation (and thus a unique product) may be found in [5, 6]. In this paper, we make no assumptions about the determinism of basic deltas or the unambiguity of delta models.

2.3 Features

For specific product lines, a set $\mathcal{F}$ of relevant feature labels is introduced. Eventually deltas are linked to feature labels, so we can generate a delta model for each legal combination of features. We do not describe this process in detail here, but we will describe the concept of a feature model, which will be used later:

Definition 6 (Feature Model). *A* feature model $\Phi \subseteq \mathscr{P}(\mathcal{F})$ *is a set of sets of feature labels from* $\mathcal{F}$*. Each* $F \in \Phi$ *is a set of feature labels corresponding to a valid* feature configuration*, i.e. a set of features that may be selected together.*

3. MODAL LOGIC

In this section, we recall a number of essential notions from the theory of modal logic [2]. We define the basic language, its semantics and the syntactic notion of a proof in general terms. Following this, in the next section, we will instantiate this theory with a language in which the modalities correspond to the deltas from our underlying abstract delta modeling framework.

3.1 Language and Semantics

We will be concerned with a basic multi-modal language in which we have a set of proposition letters, and a set of labeled modalities. In order to keep the story simple and accessible, we will only concern ourselves with unary modalities for now. In principle, however, modalities can have any arity. This basic modal language consists of the following terms:

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

Here, $\langle d \rangle$ is any unary modality labeled with d and p is any proposition letter taken from a set Ξ of proposition letters.

A frame $\mathfrak{F} = (W, R_{d_1}, \ldots)$ over this language consists of a set W of worlds and for each unary modality $\langle d \rangle$, a binary relation $R_d \subseteq W \times W$.

A model $\mathfrak{M} = (\mathfrak{F}, V)$ consists of a frame $\mathfrak{F}$ and a valuation function $V : \Xi \to \mathscr{P}(W)$, mapping proposition letters to sets of worlds. We can now, given a model $\mathfrak{M}$ and world $w \in W$, define the modal satisfaction relation $\vDash$ as follows:

$\mathfrak{M}, w \vDash \bot$	never	
$\mathfrak{M}, w \vDash p$	iff	$w \in V(p)$
$\mathfrak{M}, w \vDash \phi \vee \psi$	iff	$\mathfrak{M}, w \vDash \phi$, or $\mathfrak{M}, w \vDash \psi$
$\mathfrak{M}, w \vDash \neg\phi$	iff	not $\mathfrak{M}, w \vDash \phi$
$\mathfrak{M}, w \vDash \langle d \rangle\, \phi$	iff	there exists a $v \in W$ with $(w, v) \in R_d$ and $\mathfrak{M}, v \vDash \phi$

We regard $\top$, $\phi \wedge \psi$, $\phi \to \psi$ and $[d]\,\phi$ as abbreviations for $\neg\bot$, $\neg(\neg\phi \vee \neg\psi)$, $\neg\phi \vee \psi$ and $\neg\,\langle d\rangle\,\neg\phi$, respectively.

We furthermore write $\mathfrak{M} \vDash \phi$ and say that ϕ is *globally true in* $\mathfrak{M}$ iff for all worlds w, we have $\mathfrak{M}, w \vDash \phi$. Given a frame $\mathfrak{F}$, we write $\mathfrak{F}, w \vDash \phi$ and say ϕ is valid at world w iff for all models $\mathfrak{M}$ based on $\mathfrak{F}$, we have $\mathfrak{M}, w \vDash \phi$. We furthermore write $\mathfrak{F} \vDash \phi$ and say ϕ is valid on $\mathfrak{F}$ iff for all worlds w, we have $\mathfrak{F}, w \vDash \phi$. When we want to restrict the semantic entailment to a certain class of structures S, we subscribe $\vDash$ with S, as in $\vDash_{\mathsf{S}}$.

Given a set of formulas Γ and a class of structures S (either models or frames), we say that ϕ is a *local consequence* of Γ, and write $\Gamma \vDash_{\mathsf{S}} \phi$, iff, for all models $\mathfrak{M}$ (possibly based on frames) from S, and all worlds $w \in W$:

$$\text{if} \qquad \mathfrak{M}, w \vDash \Gamma \qquad \text{then} \qquad \mathfrak{M}, w \vDash \phi.$$

Likewise, given a set of formulas Γ and a class of structures S, we say ϕ is a *global consequence* of Γ, and write $\Gamma \vDash^{g}_{\mathsf{S}} \phi$, iff, for all models $\mathfrak{M}$ from S, we have

$$\text{if} \qquad \mathfrak{M} \vDash \Gamma \qquad \text{then} \qquad \mathfrak{M} \vDash \phi.$$

3.2 Proof Theory

We now formalize the notion of a normal modal logic, axiomatizations for them and completeness over a class of frames.

Definition 7 (Normal Modal Logic). *Given any modal language, a* normal modal logic *is a set of formulas Λ containing all propositional tautologies, the formula* **K***:*

$$[d]\,(p \to q) \to ([d]\,p \to [d]\,q),$$

the formula **Dual***:*

$$\langle d\rangle\, p \leftrightarrow \neg\,[d]\,\neg p$$

(for all modalities d), and which is closed under:

- Modus ponens*: if $\phi \in \Lambda$ and $\phi \to \psi \in \Lambda$, then $\psi \in \Lambda$;*
- Uniform substitution*: if $\phi \in \Lambda$, then $\phi[\psi/p] \in \Lambda$ for all proposition letters p and formulas ϕ, ψ; and*
- Generalization*: if $\phi \in \Lambda$, then $[d]\,\phi \in \Lambda$ for all modalities $[d]$.*

Given any set of formulas Γ, a smallest normal modal logic containing all formulas in Γ always exists, and is called the normal modal logic *generated by* Γ.

Given a normal modal logic Λ, we write

$$\vdash_{\Lambda} \phi$$

to denote $\phi \in \Lambda$, and

$$\Gamma \vdash_{\Lambda} \phi$$

to express that there are formulas $\psi_1, \ldots, \psi_n \in \Gamma$ such that

$$\vdash_{\Lambda} (\bigwedge_{1 \le i \le n} \psi_i) \to \phi.$$

Alternatively, we can also regard the relation $\vdash$ in terms of a proof system. Here, we regard **K** and **Dual**, together with all propositional tautologies as axioms, and regard the earlier closure properties (modus ponens, uniform substitution, and generalization) as proof rules.

A normal modal logic Λ is called *strongly complete* with respect to a class S of frames, if for any set of formulas Γ and any formula ϕ, $\Gamma \vDash_{\mathsf{S}} \phi$ implies $\Gamma \vdash_{\Lambda} \phi$. The normal modal logic **K**, generated by the empty set, is strongly complete with respect to the class of all frames [2].

4. DELTA FRAMES

One of the primary goals of this paper is to reason about abstract delta modeling using the language and techniques of modal logic. A good starting point, before moving on to an axiomatic characterization (in which we are concerned with issues such as completeness), is to describe delta modeling using Kripke frames.

4.1 Relational Deltas

For the convenience of the formalism described in the remainder of the paper, we now start working in a more concrete deltoid, in which deltas are relations between products.

Definition 8 (Relational Deltoid). *A* relational deltoid $(\mathcal{P}, \mathcal{D}, \cdot, \epsilon, -(-))$ *is a deltoid in which $\mathcal{D} = \mathscr{P}(\mathcal{P} \times \mathcal{P})$.*

For a complete characterization of the deltoid and a solid link to earlier work [5, 6], we also need to define delta action (Definition 2) concretely, but this is quite straightforward.

Definition 9 (Relational Delta Action). *A* relational delta action *is an operation $-(-) : \mathcal{D} \times \mathcal{P} \to \mathscr{P}(\mathcal{P})$ such that for all $x \in \mathcal{D}$ and all $p \in \mathcal{P}$:*

$$x(p) \stackrel{\text{def}}{=} \{\, q \in \mathcal{P} \mid (p, q) \in x \,\}$$

This implicitly defines sequential composition $\cdot$ as relation composition and the empty delta ϵ as the identity relation.

The paper loses no generality with this approach. The only real difference is that we lose the possibility of having distinct deltas representing the same relation.

4.2 Delta Terms

We define the set of *delta terms* (which can be seen as the syntactic counterparts of deltas) as the smallest set such that:

1. Every delta has a corresponding basic delta term d,
2. Given delta terms d_1 and d_2, $d_2 \cdot d_1$ and $d_1 \cup d_2$ are delta terms, and
3. Given a finite set D of delta terms, and a partial order $\prec : D \times D$, $(D, \prec)$ is a delta term.

From here onward, we use the set of delta terms to label our set of unary modalities. i.e. for each delta term d, there exist unary modalities $\langle d\rangle$ and $[d]$.

4.3 Frames and Relations

A relational deltoid uniquely defines a delta frame $\mathfrak{F} = (W, R_{d_1}, \ldots)$. The set of worlds W is the set of products $\mathcal{P}$ and each binary relation R_{d_i} is a delta from $\mathcal{D}$.

Definition 10. *The relation R_d is the delta corresponding to basic delta term d. We define the binary relations corresponding to compound delta terms inductively, in terms of basic delta terms. First, union and composition:*

$$\begin{aligned} R_{d_1 \cup d_2} &\stackrel{\text{def}}{=} R_{d_1} \cup R_{d_2} \\ R_{d_2 \cdot d_1} &\stackrel{\text{def}}{=} \{(p_3, p_1) \mid \exists p_2 : ((p_3, p_2) \in R_{d_2} \wedge (p_2, p_1) \in R_{d_1})\} \end{aligned}$$

Finally, the binary relation corresponding to a partial order $(D, \prec)$ on delta terms can be described in terms of derivations of this partial order as follows:

$$R_{(D,\prec)} \stackrel{\text{def}}{=} \bigcup_{d \in derv((D,\prec))} R_d,$$

Using *derv* (Definition 5) here is a bit of an abuse of notation, as it is defined on deltas, not delta terms. However, a delta term version can be defined analogously. Note that if the relations corresponding to the delta terms in D are deterministic (functional) and the partial order $(D, \prec)$ has a unique derivation, the relation $R_{(D,\prec)}$ is deterministic as well. Note also that we can characterize composition in terms of partial orders:

$$R_{d_2 \cdot d_1} = R_{(\{d_1,d_2\},\{(d_1,d_2)\})}$$

as, conversely, we can characterize partial orders in terms of union and composition.

Definition 11 (Delta frames). *Let* $\Delta\mathsf{F}$*, the class of* delta frames*, be the class of all frames, with an underlying set of delta terms as modalities, satisfying the relational equalities from Definition 10.*

Figure 1 shows part of an infinite delta frame.

We now introduce the following useful notation:

Notation 1. *For a given partially ordered set* $DM = (D, \prec)$ *and subset* $D' \subseteq D$*, we define the notation:*

$$DM \setminus D' \stackrel{\text{def}}{=} (D \setminus D', \prec')$$

where $\prec'$ *is* $\prec$ *restricted to* $D \setminus D'$*.*

Then, from Definition 10, the following proposition follows straightforwardly:

Theorem 1. *Given a nonempty delta model* $DM = (D, \prec)$ *and any formula* ϕ*, we have*

$$\models_{\Delta\mathsf{F}} \langle DM \rangle \phi \leftrightarrow \bigvee_{d \text{ minimal}} \langle d \rangle \langle DM \setminus \{d\} \rangle \phi$$

and for the empty delta model $(\varnothing, \varnothing)$*, we have*

$$\models_{\Delta\mathsf{F}} \langle (\varnothing, \varnothing) \rangle \phi \leftrightarrow \phi$$

Proof. Induction on the size of D. □

Dually, for nonempty DM, the semantic entailment

$$\models_{\Delta\mathsf{F}} [DM] \phi \leftrightarrow \bigwedge_{d \text{ minimal}} [d] [DM \setminus \{d\}] \phi$$

is also valid as a direct consequence of Theorem 1.

It is worthwhile to note that the above theorem is similar to what is known as the *expansion law* of the process algebra CCS [11]. Because delta models are finite and do not contain cycles the expansion law in combination with other axioms allows a complete reduction to basic delta terms, as explained in more detail below.

4.4 Completeness

We now discover that the normal modal logic generated by the formulas from Theorem 1 (together with axioms for union and composition) is strongly complete with respect to the class of delta frames.

Definition 12. *Define the modal logic* $\mathbf{K\Delta}$ *as the normal modal logic generated by the following axiom schemata:*

1. $\langle d_2 \cdot d_1 \rangle \phi \leftrightarrow \langle d_1 \rangle \langle d_2 \rangle \phi$*;*
2. $\langle d_1 \cup d_2 \rangle \phi \leftrightarrow (\langle d_1 \rangle \phi \vee \langle d_2 \rangle \phi)$*;*
3. $\langle DM \rangle \phi \leftrightarrow \bigvee_{d \text{ min.}} \langle d \rangle \langle DM \setminus \{d\} \rangle \phi$*; (nonempty* DM*)*
4. $\langle (\varnothing, \varnothing) \rangle \phi \leftrightarrow \phi$*.*

We call the instances of these axiom schemata '$\mathbf{\Delta}$ axioms'. They allow us to formulate the following completeness result, after defining a translation function t as follows (that t is well-defined trivially follows from defining a fitting complexity function on formulas):

Definition 13.

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

The idea behind this function is to translate any formula into an equivalent formula in which all unary modalities are labeled only by basic delta terms. This enables us to forget about compound delta terms, and to construct our completeness proof in terms of the completeness of $\mathbf{K}$ with regard to the class of all frames.

Lemma 1. *For all* Γ *and* ϕ*, we have:*

1. $\Gamma \models_{\Delta\mathsf{F}} \phi$ *iff* $\Gamma \models_{\Delta\mathsf{F}} t(\phi)$*;*
2. $\Gamma \vdash_{\mathbf{K\Delta}} \phi$ *iff* $\Gamma \vdash_{\mathbf{K\Delta}} t(\phi)$*; and*
3. $\Gamma \models_{\Delta\mathsf{F}} t(\phi)$ *iff* $\Gamma \models t(\phi)$

Proof. The first and second part of the lemma can be proved by induction (on the complexity of formulas as well as that of delta terms); the third part follows from the observation that for any translated formula, only the relations corresponding to basic delta terms are used: hence, we are simply treating our delta frame as a regular frame. □

Theorem 2. $\mathbf{K\Delta}$ *is strongly complete with regard to the class of delta frames.*

Proof. This amounts to saying that, for any Γ and ϕ, if $\Gamma \models_{\Delta\mathsf{F}} \phi$, then $\Gamma \vdash_{\mathbf{K\Delta}} \phi$. If $\Gamma \models_{\Delta\mathsf{F}} \phi$ then, by Lemma 1.1, we have $\Gamma \models_{\Delta\mathsf{F}} t(\phi)$ and by Lemma 1.3, we have $\Gamma \models t(\phi)$. Completeness of $\mathbf{K}$ now gives $\Gamma \vdash_{\mathbf{K}} t(\phi)$ and, because $\mathbf{K} \subseteq \mathbf{K\Delta}$, we also get $\Gamma \vdash_{\mathbf{K\Delta}} t(\phi)$. Finally, Lemma 1.2 yields $\Gamma \vdash_{\mathbf{K\Delta}} \phi$. □

It is possible to extend this completeness result in simple and straightforward ways. As an example of such an extension, consider the class of deterministic delta frames in which all deltas are deterministic (functional). This class of frames can be characterized by the axiom schema

$$\langle d \rangle \phi \rightarrow [d] \phi$$

for all delta terms d and all formulas ϕ. We call the normal modal logic generated by that axiom schema $\mathbf{K\Delta d}$.

Theorem 3. *The logic* $\mathbf{K\Delta d}$ *is strongly complete with regard to the class of deterministic delta frames.*

5. MODELS ON DELTA FRAMES

As we can now reason on the frame level with the proof system of Section 4, we would also like to reason on the level of models.

Recall that a model $\mathfrak{M} = (\mathfrak{F}, V)$ is a frame augmented with a valuation function $V : \Xi \to \mathscr{P}(W)$ which maps proposition letters to the set of worlds in which they are true. Our worlds are products from $\mathcal{P}$. What we want to reason about is the *features* that are implemented by those products, so we now decide that $\mathcal{F} \subseteq \Xi$. We would like to prove properties about the effect of deltas on specific features.

5.1 Semantic Feature Model

In Definition 6 we see features as labels. A feature model Φ indicates which features are allowed to be selected together on a conceptual level. However, if we have $\mathfrak{M}, w \vDash_{\Delta\mathsf{F}} f$ for some $f \in \mathcal{F}$, it means that feature f is actually implemented in product w. It is a semantic judgment.

An interesting relation exists, however, that ties the two notions together. A (syntactic) feature model is only sensible if all of its feature configurations can actually be implemented. We define a semantic feature model as follows:

Definition 14 (Semantic Feature Model). *Given a model $\mathfrak{M} = (W, R_{d_1}, \ldots, V)$, we define its semantic feature model $\Phi_{\mathfrak{M}} \subseteq \mathscr{P}(\mathcal{F})$ as the set of sets of features that can semantically be implemented together:*

$$\Phi_{\mathfrak{M}} \stackrel{\text{def}}{=} \{\, V'(w) \cap \mathcal{F} \mid w \in W \,\}$$

where $V' : W \to \mathscr{P}(\Xi)$ is the function mapping each world to the set of proposition letters that are true there:

$$V'(w) \stackrel{\text{def}}{=} \{\, p \in \Xi \mid w \in V(p) \,\}$$

We expect a sensible syntactic feature model to be a subset of the semantic feature model:

$$\Phi \subseteq \Phi_{\mathfrak{M}}$$

meaning that all valid feature configurations contain only features that can (potentially) be implemented together.

5.2 Proof System Soundness

Note that the proof system from Section 4 is not sound with respect to global semantic entailment on models. For example, consider the following 'proof':

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

So we have

$$f \to [d]\, g \;\vdash_{\mathbf{K}}\; f \to [d]\, \neg g,$$

but at the same time the (global) semantic consequence

$$f \to [d]\, g \;\vDash^{g}\; f \to [d]\, \neg g$$

is easily seen to be false. The culprit here is our use of uniform substitution. The initial axiom in our false proof is not a tautology that is 'true for all g'. It is meant as a statement about the feature g specifically. We still need the uniform substitution rule, however, to prove such truths as:

(1)	$p \vee \neg p$	propositional tautology
(2)	$[d]\, f \vee \neg\, [d]\, f$	uniform substitution on p

The trick is to allow uniform substitution only on newly produced proposition-letters, but not on the original features in our axioms. This may be accomplished by first transforming all propositions in our axioms and formulas to *nullary modalities* [3], on which uniform substitution does not apply. We can then prove valid formulas in the proof system of frames.

So we now introduce nullary modalities, which may be seen as propositional constants, into our language. A nullary modality labeled with feature f is denoted $\textcircled{f}$ and it corresponds to a unary relation U_f in our frames. It has the following meaning by our modal satisfaction relation $\vDash$:

$$\mathfrak{M}, w \vDash \textcircled{f} \quad \text{iff} \quad w \in U_f$$

We then define the following translation:

Definition 15.

$$\begin{array}{lcl} u(f) & \stackrel{\text{def}}{=} & \textcircled{f} \quad \textit{for proposition letters } f \\ u(\neg\phi) & \stackrel{\text{def}}{=} & \neg\, u(\phi) \\ \vdots & & \end{array}$$

For the other shapes of formulas the u translation is simply propagated down to the proposition letters, leaving everything else unchanged. We also lift the function u to sets of formulas in the expected manner.

We furthermore define a translation function from models to frames (overloading the earlier name u):

Definition 16. *For all $\mathfrak{M} = (W, R_{d_1}, \ldots, V)$, we define translation function u as follows:*

$$u(\mathfrak{M}) \stackrel{\text{def}}{=} (W, U_{f_1}, \ldots, R_{d_1}, \ldots)$$

where for all $f_i \in \Xi$ and all $w \in V(f_i)$ we have $w \in U_{f_i}$.

We then formulate the following translation lemma:

Lemma 2. *For all models $\mathfrak{M}$, worlds w and sets of formulas Γ, we have:*

1. $\mathfrak{M}, w \vDash \Gamma$ *iff* $u(\mathfrak{M}), w \vDash u(\Gamma)$
2. $\mathfrak{M} \vDash \Gamma$ *iff* $u(\mathfrak{M}) \vDash u(\Gamma)$

Proof. Proof of the first part of the lemma is by induction on the complexity of (sets of) formulas. The base case trivially follows from our construction of nullary modalities in terms of proposition letters. The second part of the lemma follows trivially from the first. □

This lemma enables us to prove the following soundness result with regard to global truth on the model level:

Theorem 4. *For all sets of formulas Γ and all formulas ϕ:*

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

Proof. Assume $u(\Gamma) \vdash_{\mathbf{K}\Delta} u(\phi)$. Let $\mathfrak{M}$ be a model (based on a delta frame) such that $\mathfrak{M} \vDash \Gamma$. Then, by Lemma 2.2, we have $u(\mathfrak{M}) \vDash u(\Gamma)$. Now let Λ be the logic of the class of delta frames

$$\{\mathfrak{F} \in \Delta\mathsf{F} \mid \mathfrak{F} \vDash u(\Gamma)\}.$$

Because Λ is a normal modal logic, it is closed under proof rules, and hence it follows from $u(\Gamma) \vdash_{\mathbf{K}\Delta} u(\phi)$ combined with the fact that $u(\Gamma) \subseteq \Lambda$, that $u(\phi) \in \Lambda$. It follows that $u(\phi)$ is valid on this class of frames, so we have:

$$u(\mathfrak{M}) \vDash u(\phi).$$

Lemma 2.2 now gives us $\mathfrak{M} \vDash \phi$ and hence $\Gamma \vDash^{g}_{\Delta\mathsf{F}} \phi$. □

5.3 Relative Completeness

In Hoare logic *relative* completeness has been established for classes of models which allow the expressiveness in the logic of (liberal) weakest preconditions [1]. For example in [8] a class of arithmetic models has been introduced which allow expressiveness in the logic of weakest preconditions by means of arithmetically based encoding techniques. Following this general approach to completeness of Hoare logics we want to identify a class of models for which the converse of the above Theorem 4 holds. More specifically, we want to identify a set of models $\mathfrak{M}$ for which there exists an axiomatization $\Gamma_{\mathfrak{M}}$ in $\mathbf{K\Delta}$ such that $\mathfrak{M} \vDash \phi$ implies $u(\Gamma_{\mathfrak{M}}) \vdash_{\mathbf{K}} u(\phi)$. Note that in our modal logic $\mathbf{K\Delta}$ the liberal weakest precondition of a delta d and postcondition ϕ can be directly expressed by the formula $[d]\,\phi$. A natural set of models to consider are those which allow the expression of such weakest preconditions only in terms of a logical combination of proposition letters.

Definition 17 (Precondition Expressibility). *A model $\mathfrak{M}$ allows the expression of weakest preconditions iff for every formula $[d]\,\phi$, where d is a basic delta term and ϕ is a propositional formula, there exists another propositional formula ϕ' such that for all worlds $w \in W$:*

$$\mathfrak{M}, w \vDash [d]\,\phi \qquad \textit{iff} \qquad \mathfrak{M}, w \vDash \phi'$$

Lemma 3. *In every model $\mathfrak{M}$ which allows the expression of weakest preconditions, it is also true that for every formula ϕ, there exists a propositional formula ϕ' such that for all worlds $w \in W$:*

$$\mathfrak{M}, w \vDash \phi \qquad \textit{iff} \qquad \mathfrak{M}, w \vDash \phi'$$

Moreover, this result can be lifted over the translation function u, i.e.:

$$u(\mathfrak{M}), w \vDash u(\phi) \qquad \textit{iff} \qquad u(\mathfrak{M}), w \vDash u(\phi')$$

Proof. Use translation function t to restrict the modalities in ϕ to basic delta terms, using Lemma 1. Then use induction on the nesting depth of the modalities in $t(\phi)$. □

For any model $\mathfrak{M}$ let $\Gamma_{\mathfrak{M}}$ denote its propositional theory

$$\{\phi \mid \mathfrak{M} \vDash \phi\} \qquad (\phi \text{ is propositional})$$

extended with the set of formulas

$$\{\, [d]\,\phi \leftrightarrow \phi' \mid \mathfrak{M} \vDash [d]\,\phi \leftrightarrow \phi' \,\}$$

where d is a basic delta term, and both ϕ and ϕ' are propositional formulas. We have the following relative completeness theorem.

Theorem 5. *For any model $\mathfrak{M}$ that allows the expression of weakest preconditions we have, for all formulas ϕ:*

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

Proof. Assume $\mathfrak{M} \vDash \phi$. By Lemma 3 there is a propositional formula ϕ' equivalent to ϕ such that $\mathfrak{M} \vDash \phi'$.

This gives us $\phi' \in \Gamma_{\mathfrak{M}}$, which can be lifted to

$$u(\phi') \in u(\Gamma_{\mathfrak{M}})$$

from which $u(\Gamma_{\mathfrak{M}}) \vdash_{\mathbf{K\Delta}} u(\phi')$ follows directly.

To complete the proof, we only need to show that

$$u(\Gamma_{\mathfrak{M}}) \vdash_{\mathbf{K\Delta}} u(\phi') \to u(\phi)$$

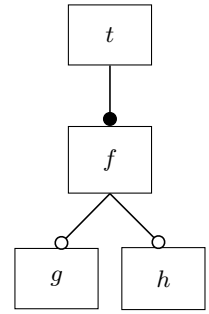

Figure 2: Example feature model

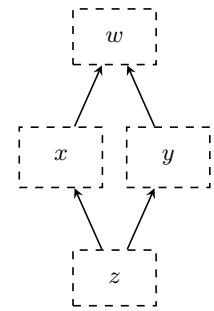

Figure 3: Example delta model *DM*

but this follows from the fact that ϕ can be constructed from ϕ' inductively (and, likewise, $u(\phi)$ from $u(\phi')$) using the equivalences between basic modal formulas and propositional formulas, which are included in $\Gamma_{\mathfrak{M}}$. Finally, a simple application of modus ponens gives us the desired result. □

Note that for models that allow the expression of weakest preconditions our modal logic $\mathbf{K\Delta}$ is in fact a *conservative extension* of the propositional logic of the underlying semantic feature models. Also of interest is that for deterministic delta frames we only need to require that every formula $[d]\,f$ – where f is a single feature or feature negation – can be expressed by a propositional logic formula. This is because in deterministic frames, the box modality distributes over disjunction and the diamond modality over conjunction.

5.4 Example

We now illustrate the use of $\mathbf{K\Delta}$ through an example proof. Say we have the feature model as shown in Figure 2 [16]. The features f, g and h are implemented by the delta model *DM* in Figure 3. The feature t is satisfied in some empty core product, on which we'd like to apply those deltas.

We now introduce a set of basic axioms valid in this model:

Axiom 1 (Delta Model Axioms).

(1)	$f \to t$	(6)	$t \to [w]\,f$
(2)	$g \to f$	(7)	$f \to [x]\,g$
(3)	$h \to f$	(8)	$f \to [y]\,h$
(4)	$g \to [y]\,g$	(9)	$g \to [z]\,g$
(5)	$h \to [x]\,h$	(10)	$h \to [z]\,h$

1, 2 and 3 are due to the feature model shown in Figure 2. It is generally the case that when a subfeature is implemented its superfeature is implemented as well. 4 and 5 are due to a property we assume the underlying deltoid to have, called non-interference [9], which states that commuting deltas cannot interfere with each others features. 6 to 10 are by design of the deltas w, x, y and z. We assume that they were developed such that w, x and y implement the features f, g and h (6, 7 and 8), taking into account only the deltas 'above' them, and that conflict resolving delta z [5, 6] doesn't break the features implemented by the previous deltas (9 and 10).

Axioms 6 to 10 are enforced by the developers of the product line when they follow the Delta Modeling Workflow as described in [9, 10]. The workflow ensures desirable global properties by design if local constraints such as axioms 6 to

10 are met. Now say we have a core product $c \in \mathcal{P}$ with $c \vDash t$. For our example, we'd like to prove the following global property about delta model DM:

PROPOSITION 1. $c \vDash [DM]\,(t \wedge f \wedge g \wedge h)$

In order to prove this property more succinctly, we introduce the following auxiliary proof rules:

LEMMA 4 (l4). *For all formulas ϕ, ψ and χ, and for all delta terms $d_1, \ldots, d_n$, we have:*

$$\phi \to [d_1] \cdots [d_n]\,\psi, \quad \psi \to \chi \quad \vdash \quad \phi \to [d_1] \cdots [d_n]\,\chi$$

PROOF. By induction on n. □

LEMMA 5 (l5). *For all formulas ϕ and ψ and all delta terms d, we have:*

$$\vdash \quad ([d]\,\phi \wedge [d]\,\psi) \;\leftrightarrow\; [d]\,(\phi \wedge \psi)$$

PROOF. See [2, Example 1.40]. □

In the proof of Proposition 1 shown below, we refer to the proof rules we use, as well as to previous results and axioms. When we refer to numbers 1 to 10, we actually refer to the u translation of the corresponding item from Axiom 1.

PROOF OF PROPOSITION 1.

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

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

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

where

$$\begin{aligned} DM_1 &= DM \setminus \{w, x, y\} \\ DM_2 &= DM \setminus \{w, x\} \\ DM_3 &= DM \setminus \{w, y\} \\ DM_4 &= DM \setminus \{w\} \end{aligned}$$

Then, by Theorem 4 and $c \vDash t$, we have our result. □

We have skipped many steps in this proof, mostly those concerned with invoking propositional tautologies and applying modus ponens. We have kept only the most interesting steps – those that directly use our axioms.

Partial motivation for the logic presented in this paper is to make formal proofs about the Delta Modeling Workflow more transparent. The above proof is just an example of a proof of product line completeness for a specific case. A full description of the workflow using $\mathbf{K\Delta}$, including general proofs, is planned as future work.

5.5 Alternate Propositions

In Section 5 we have chosen the set of features $\mathcal{F}$ as our significant set of propositions. But there are several reasons for choosing an alternate or additional set of propositions.

First, there may be some desired interaction between features that would not be satisfied by an implementation of any strict subset of those features. In that case, we'd want to have sets of features $\mathscr{P}(\mathcal{F}) \subseteq \Xi$ rather than individual features. We would then assume the additional axiom schema:

$$F \cup G \;\to\; F \wedge G$$

for all $F, G \subseteq \mathcal{F}$. This approach was taken in [9].

Furthermore, it is possible that different products may implement the exact same features. So we may want additional proposition letters to distinguish between them in our logic and reason on a somewhat lower level. Such proposition letters may, for example, represent the presence of specific classes or methods in an object oriented setting.

6. CONCLUSION

In this paper we provided a method that will be useful for further research into abstract delta modeling. The modal logic $\mathbf{K\Delta}$ allows us to reason more succinctly about the semantics of deltas and delta models in a way consistent with previous work. We proved strong completeness of the logic with respect to the class of all delta frames. We also discussed a proof technique on the level of models, proved its completeness and illustrated it through example.

The delta theory in this paper is based on Abstract Delta Modeling [5, 6]. We remain in a similarly abstract setting, and generalize even further by removing the assumption that deltas are deterministic and terminating entities.

Completeness proofs in modal logic have a long-standing history, closely tied to the history of relational semantics based on Kripke frames. A comprehensive survey of this history can be found in e.g. [2, Section 1.8].

The modal logic presented in this paper has a flavour very reminiscent of dynamic logics such as PDL [2, 7]. A crucial difference, however, is that the logic presented here is simpler (and hence, easier to work with) due to the absence of operations such as iteration or tests. Due to this simplicity, we can easily unravel complex modalities into simpler ones, and under certain conditions even reduce them to propositional formulas, enabling us to obtain the main results from Sections 4 and 5.

Possible future work following up the initial research in this paper may include work on characterizations of modal expressiveness of basic properties of delta models and interactions between deltas, including positive as well as limitative results. In the case of limitative results, it may be worthwhile to look into the additional expressiveness that the modal μ-calculus has to offer [4]. This additional expressiveness may, for example, be required to express the condition that a conflict between two deltas is resolved by a third delta, a property described in [5, 6].

Another interesting research direction is the use of our logical framework in the synthesis of delta models using model checking techniques.

The modal logic, as it stands now, will be used to simplify and extend the Delta Modeling Workflow formalism [9, 10], and to prove preservation of desirable properties (such as unambiguity and feature completeness) in the general case.

7. REFERENCES

[1] Jan A. Bergstra and J. V. Tucker. Expressiveness and the completeness of Hoare's logic. *J. Comput. Syst. Sci.*, 25(3):267–284, 1982.

[2] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge University Press, 2001.

[3] Patrick Blackburn, Johan F. A. K. van Benthem, and Frank Wolter. *Handbook of Modal Logic, Volume 3 (Studies in Logic and Practical Reasoning)*. Elsevier Science Inc., 2006.

[4] Julian Bradfield and Colin Stirling. *Modal mu-Calculi*, pages 721–756. In [3], 2001.

[5] D. Clarke, M. Helvensteijn, and I. Schaefer. Abstract delta modeling. In *Proc. of GPCE*, pages 13–22. ACM, 2010.

[6] D. Clarke, M. Helvensteijn, and I. Schaefer. Abstract delta modeling. *Accepted to MSCS special issue*, 2012.

[7] M. J. Fischer and R. E. Landner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18(2):194–211, 1979.

[8] David Harel. Arithmetical completeness in logics of programs. In *ICALP*, pages 268–288, 1978.

[9] M. Helvensteijn. Delta Modeling Workflow. In *Proceedings of the 6th International Workshop on Variability Modelling of Software-intensive Systems, Leipzig, Germany, January 25-27 2012*, ACM International Conference Proceedings Series. ACM, 2012.

[10] M. Helvensteijn, R. Muschevici, and P.Y.H. Wong. Delta Modeling in Practice, a Fredhopper Case Study. In *Proceedings of the 6th International Workshop on Variability Modelling of Software-intensive Systems, Leipzig, Germany, January 25-27 2012*, ACM International Conference Proceedings Series. ACM, 2012.

[11] Robin Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer, 1980.

[12] K. Pohl, G. Böckle, and F. Van Der Linden. *Software Product Line Engineering: Foundations, Principles, and Techniques*. Springer, Heidelberg, 2005.

[13] I. Schaefer. Variability Modelling for Model-Driven Development of Software Product Lines. In *Intl. Workshop on Variability Modelling of Software-intensive Systems (VaMoS 2010)*, 2010.

[14] I. Schaefer, L. Bettini, V. Bono, F. Damiani, and N. Tanzarella. Delta-oriented Programming of Software Product Lines. In *SPLC*, volume 6287 of *LNCS*, pages 77–91. Springer, 2010.

[15] I. Schaefer, A. Worret, and A. Poetzsch-Heffter. A Model-Based Framework for Automated Product Derivation. In *Proc. of Workshop in Model-based Approaches for Product Line Engineering (MAPLE 2009)*, 2009.

[16] P. Schobbens, P. Heymans, and J. Trigaux. Feature diagrams: A survey and a formal semantics. *Requirements Engineering, IEEE International Conference on*, 0:139–148, 2006.