$\mathcal{M}$odular-$\mathcal{E}$: An Elaboration Tolerant Approach to the Ramification and Qualification Problems

Antonis Kakas[1], Loizos Michael[2], and Rob Miller[3]

[1] University of Cyprus, P. O. Box 20537, CY-1678, Cyprus
antonis@ucy.ac.cy
[2] Harvard University, Cambridge, MA 02138, USA
loizos@eecs.harvard.edu
[3] University College London, London WC1E 6BT, UK
rsm@ucl.ac.uk

Abstract. We describe $\mathcal{M}$odular-$\mathcal{E}$ ($\mathcal{ME}$), a specialized, model-theoretic logic for narrative reasoning about actions, able to represent non-deterministic domains involving concurrency, static laws (constraints) and indirect effects (ramifications). We give formal results which characterize $\mathcal{ME}$'s high degree of modularity and elaboration tolerance, and show how these properties help to separate out, and provide a principled solutions to, the endogenous and exogenous qualification problems. We also show how a notion of (micro) processes can be used to facilitate reasoning at the dual levels of temporal granularity necessary for narrative-based domains involving "instantaneous" series of indirect and knock-on effects.

1 Introduction

Domain descriptions for reasoning about actions and change (RAC) in commonsense reasoning and other contexts should be *Elaboration Tolerant* [10,9]. Formalisms should be able to incorporate new information gracefully into representations, e.g. by the simple addition of sentences. Elaboration Tolerance (ET) is strongly linked with the need to have a *modular* semantics for RAC frameworks that properly separates different aspects of the domain knowledge, as argued e.g. in [6]. ET and modularity are known to be strongly related to the *Qualification Problem* in RAC – if the effect laws (or action executability laws) of our domain are not qualified in a complete way they can lead to unintended conclusions that contradict new information. In particular, new narrative information about observations or attempted actions can render the domain description inconsistent in this way.

In this paper, we present the language $\mathcal{M}$odular-$\mathcal{E}$ ($\mathcal{ME}$) as a case study in developing modular semantics for RAC frameworks in order to provide comprehensive solutions to the ramification and qualification problems. Our approach builds upon [7] and is inspired by [13], separating out the qualification problem into two parts - an *endogenous* aspect concerning qualifications expressible in the known domain language, and an *exogenous* aspect where change is qualified by

C. Baral et al. (Eds.): LPNMR 2005, LNAI 3662, pp. 211–226, 2005.

unrepresented (or exogenous) factors. The semantics of $\mathcal{ME}$ decouples these two problems, allowing exogenous qualifications to come into play only when the endogenous qualification alone is not sufficient to avoid inconsistency. It uses a simple default minimization of exogenous qualifications to "minimize *unexplained* failure" (c.f. [13]) when observations of properties cannot be reconciled with the assumed success of the applied effect laws. $\mathcal{ME}$'s modular semantics offers a clean solution to the problem of anomalous models that arose from earlier treatments of the qualification problem.

To achieve the semantic decoupling of endogenous and exogenous qualifications it is important to address two issues. First, a proper treatment of ramifications, including non-determinism and loops in chains of instantaneous effects, is needed (as any incomplete treatment will cause some endogenous qualifications to be treated as exogenous). $\mathcal{ME}$ uses a notion of *processes* for this. Second, for the same reason a full account is needed for the qualifications that static constraints provide for causal laws. In this regard we distinguish between *local or explicit* and *global or implicit* qualification. Local qualifications are the explicit preconditions included in individual causal effect laws and action executability statements. Global qualifications are formed at the semantic level by taking into account static laws and interactions between effect laws. Global qualification is closely related to modularity. Without it elaboration tolerance is compromised by the need to manually reconcile each local set of qualifications with each new static law.

We show that this analysis of the qualification and ramification problems indeed results in modularity and elaboration tolerance. For example, $\mathcal{ME}$ enjoys a "free will" property – a domain description can be extended with any action attempt at any time after its recorded observations without affecting the conclusions about the domain up to that time.

2 $\mathcal{ME}$ Syntax and Examples

In this section we give $\mathcal{ME}$'s syntax and sketch its important characteristics via a series of examples.

Definition 1 (Domain Language). *An $\mathcal{ME}$ **domain language** is a tuple $\langle \Pi, \preceq, \Delta, \Phi \rangle$, where $\preceq$ is a total ordering defined over the non-empty set Π of time-points, Δ is a non-empty set of action constants, and Φ is a non-empty set of fluent constants.*

Definition 2 (Formula, Literal and Conjunction). *A **fluent formula** is a propositional formula containing only fluent constants (used as extra-logical symbols), the standard connectives $\neg$, $\rightarrow$, $\leftarrow$, $\leftrightarrow$, $\vee$ and $\wedge$, and the truth value constants $\top$ and $\bot$. A **fluent literal** is either a fluent constant or its negation. An **action literal** is either an action constant or its negation. A **fluent conjunction** is a conjunction of fluent literals.*

Definition 3 (Converse). *Let E be an action or fluent constant. The **converse** of E, written $\overline{E}$, is $\neg E$, and the converse of $\neg E$, written $\overline{\neg E}$, is E.*

Definition 4 (Domain Description or Theory). *A* ***domain description or theory*** *in $\mathcal{ME}$ is a collection of the following types of statements, where ϕ is a fluent formula, T is a time point (assume an integer or real number unless otherwise stated), A is an action constant, C is a (possibly empty) set of fluent and action literals, L is a fluent literal, and E is a non-empty set of action constants and fluent literals:*

- ***h-propositions*** *of the form:* ϕ `holds-at` T
- ***o-propositions*** *of the form:* A `occurs-at` T
- ***c-propositions*** *of the form:* C `causes` L
- ***p-propositions*** *of the form:* ϕ `prevents` E
- ***a-propositions*** *of the form:* `always` ϕ

A domain description is ***finite*** *if it contains only a finite number of propositions.*

Singleton sets of fluent or action literals in c-propositions of the form $\{P\}$ will sometimes be written without enclosing braces, i.e. as P.

The intended meaning of h-propositions is straightforward – they can be used to record "observations" about the domain along the time line. "A `occurs-at` T" means that an attempt to execute A occurs at T. Together, the h- and o-propositions describe the "narrative" component of a domain description. "C `causes` L" means that, at any time-point, the combination of actions, inactions and preconditions described via C will *provisionally* cause L to hold immediately afterwards. As we shall see, the provisos automatically accompanying this causal rule are crucial – in any model the potential effect L competes with other potential effects, and maybe overridden, for example, because it would otherwise result in a more-than-instantaneous violation of a domain constraint described with an a-proposition. The rule "C `causes` L" is thus qualified both locally (via C) and globally via the total set of c-, p- and a-propositions. "ϕ `prevents` E" means that the circumstances described by ϕ prevent the simultaneous causation/execution of the effects/actions listed in E. " `always` ϕ" means that $\neg\phi$ can never hold, other than in temporary, instantaneous "transition states" which form part of an instantaneous chain of indirect effects. In other words, " `always` ϕ" describes a domain constraint or static law at the granularity of observable time.

Example 1 (Lift Door). A lift door can be opened and closed by pressing the "open" and "close" buttons respectively. The door is initially open, and both buttons are pressed simultaneously. This scenario can be described with a single fluent *DoorOpen* and two actions *PressOpen* and *PressClose*:

$\{PressOpen\}$ `causes` $DoorOpen$ (LD1)

$\{PressClose\}$ `causes` $\neg DoorOpen$ (LD2)

$DoorOpen$ `holds-at` 1 (LD3)

$PressOpen$ `occurs-at` 2 (LD4)

$PressClose$ `occurs-at` 2 (LD5)

Example 1 results in two models – one in which the door is open at times after 2 and one in which the door is closed. Note that, even though the conflicting actions are not prevented from occurring together (i.e. there is no p-proposition "$\top$ `prevents` $\{PressOpen, PressClose\}$"), they do not give rise to inconsistency. More generally, we show in Section 4 that $\mathcal{ME}$ exhibits a "free will" property – from any consistent initial state, and for any given collection of c- and p-propositions, any series of actions may be attempted without giving rise to inconsistency. Put another way, any finite collection of o-, c- and p-propositions is consistent with any internally consistent collection of a-propositions. Consequently, the only way to engineer an inconsistent $\mathcal{ME}$ domain description (other than by inclusion of inconsistent a-propositions) is to include "observations" (h-propositions) along the time line which contradict the predictions that would otherwise be given by $\mathcal{ME}$'s semantics. In Section 5 we show how this remaining type of inconsistency can sometimes be overcome by attributing it to unknown exogenous reasons and applying a simple minimization to these.

The following series of "broken car" examples is to illustrate the modularity and elaboration tolerance of $\mathcal{ME}$, and how this is linked to the way a- and c-propositions interact.

Example 2 (Broken Car A). Turning the key of a car causes its engine to start running. The key is turned at time 1:

$$\{TurnKey\}\ \texttt{causes}\ Running \qquad \text{(BC1)}$$
$$TurnKey\ \texttt{occurs-at}\ 1 \qquad \text{(BC2)}$$

In all models of this domain the car engine is running at all times after 1. (A more complete description would typically include some local qualifications for (BC1), e.g. "$\{TurnKey, BatteryOK\}$ `causes` $Running$" – turning the key starts the engine only when the battery is OK, in which case models would also arise where e.g. $\neg BatteryOK$ and $\neg Running$ at all time-points.)

Example 3 (Broken Car B). We elaborate the previous description by stating that broken cars' engines cannot run:

$$\texttt{always}\ \neg(Broken \wedge Running) \qquad \text{(BC3)}$$

There are two classes of models for the elaborated domain (BC1)-(BC3) – one in which the car is broken and not running at times after 1, and one in which the car is not broken and running. The occurrence of *TurnKey* at 1 does not eliminate the model in which the car is broken because the semantics of $\mathcal{ME}$ allows (BC3) to act as a global qualification, in particular for (BC1). The *TurnKey* action does not force $\neg Broken$ at earlier times, and thus if in addition the car is known to be broken the theory remains consistent after this elaboration. Without this characteristic, we would have to alter (BC1) to "$\{TurnKey, \neg Broken\}$ `causes` $Running$" to accommodate (BC3), in other words explicitly encode as a local qualification the global qualification effect of (BC3) on (BC1). In $\mathcal{ME}$ this local qualification is redundant thus illustrating its modular nature; the a-proposition (BC3) has been simply added without further ado.

Example 4 (Broken Car C). We elaborate Example 3 with two more causal rules and an extra action occurrence:

{*Break*} `causes` *Broken* (BC4)
{*Broken*} `causes` ¬*Running* (BC5)
Break `occurs-at` 1 (BC6)

In all models of the domain (BC1)-(BC6), the car is broken and not running at times after 1. (BC5) describes an "indirect effect" or "ramification". It introduces an asymmetry between the *Running* and *Broken* fluents and their relationship with (BC3), preventing (BC3) from acting as a qualification for (BC4) in the same way as it does for (BC1). Translating global to local/explicit qualifications is therefore complex, as it requires consideration of the interactions between a- and c-propositions. $\mathcal{ME}$ deals with indirect effects by considering chains of instantaneous, temporary transition states ("nodes"). Within these causal chains, "processes" are introduced to describe the initiation and termination of fluents. These processes may "stretch" across several links of a given chain before they are complete, thus allowing all possible micro-orderings of effects to be considered. Because of the coarseness of the domain description with respect to the granularity of time, this is important for a proper treatment of collections of instantaneous effects which compete or "race" against each other. Furthermore, since the granularity of time in which these chains operate is finer than that of observable time, intermediate states within them may (temporarily) violate the static laws described by a-propositions. In Example 4, one of the chains allowed by the semantics completes the process initiating *Running* and then the process initiating *Broken*. At this point there is a state in which (BC3) is violated, but (BC5) then generates a new process terminating *Running* whose completion results in a consistent state further along the chain.

Example 5 (Broken Car B+/C+). We elaborate the previous two descriptions by observing the car running at time 2:

Running `holds-at` 2 (BC-obs)

Adding (BC-obs) to Example 3 does not result in inconsistency, but allows us to infer that the car is not broken (in particular at earlier times). Note that $\mathcal{ME}$ would facilitate the opposite conclusion (*Broken*) in exactly the same way had the observation been "¬*Running* `holds-at` 2". This is because it accords exactly the same status to globally derived qualifications (in this case from (BC3)) as to qualifications localized to particular c-propositions. However, adding (BC-obs) to Example 4 does give rise to inconsistency at the level of the $\mathcal{ME}$'s "base semantics" (as detailed in Section 3), because since there are no (local or globally derived) qualifications to (BC4) and (BC5), the theory would otherwise entail ¬*Running*. An intuitive explanation for (BC-obs) in this context is that one or both of the effects of (BC4) and (BC5) "failed" due to exogenous circumstances (i.e. factors not included in the representation) implicitly qualifying these causal rules. This type of reasoning is captured within $\mathcal{ME}$ by the use of simple default minimization of such exogenous qualifications (see Section 5). The minimization policy is straightforward and robust because the base semantics fully accounts for all endogenous qualifications (i.e. those expressed in the domain) by its modularity and its encapsulation of global as well as local qualifications, as described above.

Example 6 (Broken Car D). We elaborate Example 4 with the knowledge that the car was parked at time 0 in anti-theft mode (ATM), so that causing the engine to run (even for an instant) will trigger the alarm:

$$(\neg Broken \wedge \neg Running \wedge \neg Alarm \wedge ATM)\ \texttt{holds-at}\ 0 \qquad \text{(BC7)}$$
$$\{Running, ATM\}\ \texttt{causes}\ Alarm \qquad \text{(BC8)}$$

Intuitively, even though at times after 1 the car will be broken and not running, the alarm may or may not be triggered in this narrative, depending on whether the (indirect) effect of the *Break* action takes effect just before or just after the effect of the *TurnKey* action. This is an example of a “race” condition between competing instantaneous effects. $\mathcal{ME}$ is able to deal correctly with such representations via its processed-based semantics. It gives two models of this domain – in both models $(Broken \wedge \neg Running)$ is true at times after 1, but in one model *Alarm* is true and in the other it is false. The example illustrates how $\mathcal{ME}$'s processes operate at a finer level of temporal granularity than “observable time” in order to deal with “instantaneous” indirect effects.[1]

Example 7 (Oscillator).

$$\{On\}\ \texttt{causes}\ \neg On \qquad \text{(OSC1)}$$
$$\{\neg On\}\ \texttt{causes}\ On \qquad \text{(OSC2)}$$

This example (which might e.g. represent the internal mechanism of an electric buzzer) has an infinite number of models in which the truth value of *On* is arbitrarily assigned at each time point. It illustrates that $\mathcal{ME}$ is able to deal with “loops” of indirect effects without over-constraining models. It is important, for example, not to restrict the set of models to those in which the truth value of *On* alternates at each successive time-point. This is because the change within the domain is happening ”instantaneously” – i.e. at an altogether finer granularity of time than “observable” time. Therefore the observable time-points are best considered as arbitrarily spaced “snapshots” of the finer-grained time continuum. A full treatment of such loops along these lines (as well as a full treatment of concurrency and nondeterminism) is necessary for $\mathcal{ME}$ to exhibit the “free will” property and resulting modularity and elaboration tolerance described above.

[1] An interesting (and more contentious) variation of Example 6 is to delete (BC4) and (BC6), and replace (BC7) with “$(Broken \wedge \neg Running \wedge \neg Alarm \wedge ATM)$ `holds-at` 0”. (so that the car is already broken at 1). $\mathcal{ME}$'s semantics still gives the two models with *Alarm* true in one and false in the other. This is because it treats (BC3) only as a “stability” constraint at the temporal granularity of “observable” time, and not as a “definitional” constraint that would transcend all levels of temporal granularity. Note, however, that we could eliminate the model in which *Alarm* was true by adding the p-proposition “*Broken* `prevents` *Running*”, meaning that *Broken* prevents *Running* from being caused (even instantaneously).

3 $\mathcal{M}$odular-$\mathcal{E}$ Base Semantics

In this section we give a formal account of $\mathcal{ME}$'s semantics. We begin with some straightforward preliminary definitions concerning states and processes.

3.1 Definitions Regarding States, Processes and Causal Change

Definition 5 (States and Satisfaction). *A* ***state*** *is a set S of fluent literals such that for each fluent constant F, either $F \in S$ or $\neg F \in S$ but not both. A formula ϕ* ***is satisfied in a state*** *S iff the interpretation corresponding to S is a model of ϕ.*

Definition 6 (A-Consistency). *Let D be a domain description and S a state. S is* ***a-consistent with respect to*** *D iff for every a-proposition "*`always` *ϕ" in D, ϕ is satisfied in S. D is* ***a-consistent*** *iff there exists a state which is a-consistent with respect to D. Let D_a denote the set of all a-propositions in D. Then given a fluent formula ψ, $D_a \models_a \psi$ iff ψ is entailed classically by the theory $T = \{\phi \mid$* `always` *$\phi \in D\}$.*

Definition 7 (Process). *A* ***process*** *is an expression of the form $\uparrow F$ or $\downarrow F$, where F is a fluent constant of the language. $\uparrow F$ is called* ***the initiating process of*** *F and $\downarrow F$ is called* ***the terminating process of*** *F. The* ***associated processes*** *of the c-propositions "C* `causes` *F" and "C* `causes` *$\neg F$" are respectively $\uparrow F$ and $\downarrow F$. $\uparrow F$ and $\downarrow F$ will also sometimes be written as proc(F) and proc$(\neg F)$ respectively. An* ***active process log*** *is a set of processes.*

Definitions 8 – 15 concern the identification of fluent changes following instantaneously from a given state and set of actions. A *causal chain* represents a possible instantaneous series of knock-on effects implied by the causal laws. There is a repeated two-phase mechanism for constructing the "nodes" of causal chains – a triggering phase in which new processes are generated from c-propositions applicable at that point, immediately followed by a resolution phase in which some of the already-active processes complete, resulting in an update of the corresponding fluents' truth values. The process triggering is appropriately limited by the p-propositions. The triggering and completion of a particular process may be separated by several steps in the chain, so that consideration of all such chains gives an adequate treatment of "race" conditions between competing instantaneous effects. Chains terminate either because they reach a state from which no change is possible (a *static node*) or because they loop back on themselves. We have made the working (but retractable) assumption that actions trigger processes only at the beginning of such chains, at which point they are "consumed".

Definition 8 (Causal Node). *A* ***causal node*** *(or* ***node****) is a tuple $\langle S, B, P\rangle$, where S is a state, B is a set of action constants and P is an active process log. $\langle S, B, P\rangle$ is* ***fully resolved*** *iff $P = \emptyset$, and is* ***a-consistent w.r.t. a domain description*** *D iff S is a-consistent w.r.t. D.*

Definition 9 (Triggering). *Let D be a domain description, $N = \langle S, B, P\rangle$ a node, L_t a set of fluent literals, $P_t = \{proc(L) \mid L \in L_t\}$, and B_t a set of action constants. The set $(B_t \cup P_t)$ is* ***triggered at*** *N* ***with respect to*** *D iff*

1. *$B_t \subseteq B$*
2. *For each p-proposition "ϕ* `prevents` *E" in D, either ϕ is not satisfied in S or $E \nsubseteq (B_t \cup L_t)$.*
3. *For each $L \in L_t$ there is a c-proposition "C* `causes` *L" in D such that (i) for each action constant $A \in C$, $A \in B_t$, (ii) for each action literal $\neg A \in C$, $A \notin B_t$, and (iii) for each fluent literal $L' \in C$, $L' \in S$.*

$(B_t \cup P_t)$ is ***maximally triggered at*** *N* ***with respect to*** *D iff there is no other set $(B'_t \cup P'_t)$ also triggered at N with respect to D and $(B_t \cup P_t)$ is a strict subset of $(B'_t \cup P'_t)$.*

Definition 10 (Process Successor). *Let D be a domain description and $N = \langle S, B, P\rangle$ a node. A* ***process successor of*** *N* ***w.r.t.*** *D is a node of the form $\langle S, B_t, (P \cup P_t)\rangle$, where $(B_t \cup P_t)$ is maximally triggered at N with respect to D.*

Definition 11 (Resolvant). *Let $N = \langle S, B, P\rangle$ and $N' = \langle S', \emptyset, P'\rangle$ be causal nodes. N' is a* ***resolvant of*** *N iff $S' = S$ and $P = P' = \emptyset$ or there exists a non-empty subset R of P such that the following conditions hold.*

1. *$P' = P - R$.*
2. *For each fluent constant F such that both $\uparrow F$ and $\downarrow F$ are in P, either both or neither $\uparrow F$ and $\downarrow F$ are in R.*
3. *For each fluent constant F (i) if $\uparrow F \in R$ and $\downarrow F \notin R$ then $F \in S'$, (ii) if $\downarrow F \in R$ and $\uparrow F \notin R$ then $\neg F \in S'$, (iii) if $\downarrow F \notin R$ and $\uparrow F \notin R$ then $F \in S'$ iff $F \in S$.*

N' is a ***full resolvant of*** *N iff $P' = \emptyset$.*

Definition 12 (Stationary/Static Nodes). *Let D be a domain description and $N = \langle S, B, P\rangle$ a causal node. N is* ***stationary*** *iff for each resolvant $\langle S', \emptyset, P'\rangle$ of N, $S' = S$. N is* ***static w.r.t.*** *D iff every process successor of N w.r.t. D is stationary.*

The central definition of causal chains now follows. It is slightly complicated by the need to deal with loops – conditions 2, 3 and 4 below ensure that all chains will end when the first static or repeated node is encountered.

Definition 13 (Causal Chain). *Let D be a domain description and let N_0 be a node. A* ***causal chain rooted at*** *N_0* ***with respect to*** *D is a (finite) sequence $N_0, N_1, ..., N_{2n}$ of nodes such that for each k, $0 \leq k \leq n-1$, N_{2k+1} is a process successor of N_{2k} w.r.t. D and N_{2k+2} is a resolvant of N_{2k+1}, and such that the following conditions hold:*

1. *N_{2n} is fully resolved.*
2. *N_{2n} is static, or there exists $k < n$ s.t. $N_{2n} = N_{2k}$.*

3. If there exists $j < k \leq n$ s.t. $N_{2j} = N_{2k}$ then $k = n$.
4. There does not exist a $k < n$ s.t. N_{2k} is static.

In the context of Example 1, Figure 1 below shows the tree of all possible causal chains with the starting node $\langle\{DoorOpen\}, \{PressClose, PressOpen\}, \emptyset\rangle$ (which intuitively corresponds to the situation at time 2). N_1 is the unique process successor of N_0, and the nodes N_2 and N_2' (which are both static) are the only resolvants of N_1.

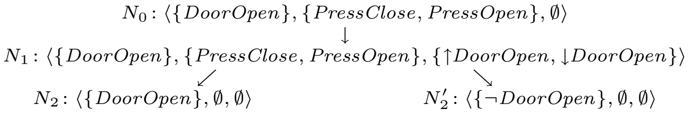
$N_0: \langle\{DoorOpen\}, \{PressClose, PressOpen\}, \emptyset\rangle$
$N_1: \langle\{DoorOpen\}, \{PressClose, PressOpen\}, \{\uparrow DoorOpen, \downarrow DoorOpen\}\rangle$
$N_2: \langle\{DoorOpen\}, \emptyset, \emptyset\rangle$
$N_2': \langle\{\neg DoorOpen\}, \emptyset, \emptyset\rangle$

Fig. 1.

As regards Example 6, we may form several causal chains starting from the node corresponding to time 1. Here is a chain terminating with a state in which *Alarm* holds ($Br = Broken$, $Ru = Running$, $Al = Alarm$):

N_0: $\langle\{\neg Br, \neg Ru, \neg Al, ATM\}, \{Break, TurnKey\}, \emptyset\rangle$
N_1: $\langle\{\neg Br, \neg Ru, \neg Al, ATM\}, \{Break, TurnKey\}, \{\uparrow Br, \uparrow Ru\}\rangle$
N_2: $\langle\{Br, Ru, \neg Al, ATM\}, \emptyset, \emptyset\rangle$
N_3: $\langle\{Br, Ru, \neg Al, ATM\}, \emptyset, \{\downarrow Ru, \uparrow Al\}\rangle$
N_4: $\langle\{Br, \neg Ru, Al, ATM\}, \emptyset, \emptyset\rangle$

Here is another chain terminating with a state in which $\neg Alarm$ holds:

N_0: $\langle\{\neg Br, \neg Ru, \neg Al, ATM\}, \{Break, TurnKey\}, \emptyset\rangle$
N_1: $\langle\{\neg Br, \neg Ru, \neg Al, ATM\}, \{Break, TurnKey\}, \{\uparrow Br, \uparrow Ru\}\rangle$
N_2': $\langle\{Br, \neg Ru, \neg Al, ATM\}, \emptyset, \{\uparrow Ru\}\rangle$
N_3': $\langle\{Br, \neg Ru, \neg Al, ATM\}, \emptyset, \{\downarrow Ru, \uparrow Ru\}\rangle$
N_4': $\langle\{Br, \neg Ru, \neg Al, ATM\}, \emptyset, \emptyset\rangle$

Nodes, and in particular nodes that terminate causal chains, do not necessarily contain a-consistent states. But causal chains that do not terminate a-consistently are not discarded when computing direct and indirect instantaneous effects. Rather, the semantics identifies *proper causal descendants* within a tree of all possible causal chains starting from a given root node. These are a-consistent nodes which are either within the terminating loop of a chain (condition 1 in Definition 14), or are such that there are no other a-consistent nodes further from the root of the tree (condition 2). (For example, in Fig. 1, N_2 and N_2' are proper causal descendants of N_0 by condition 1 below, with $j = k = n = 1$.)

Definition 14 (Proper Causal Descendant). *Let D be a domain description and let N_0 and N be nodes. N is a* ***proper causal descendant of*** *N_0* ***w.r.t.***

D iff N is a-consistent w.r.t. D, and there exists a causal chain $N_0, N_1, ..., N_{2n}$ w.r.t. D such that $N = N_{2k}$ for some $0 \leq k \leq n$ and at least one of the following two conditions holds:

1. *There exists $j \leq k$ such that $N_{2j} = N_{2n}$.*
2. *There does not exist a causal chain $N_0, N_1, ..., N_{2k}, N'_{2k+1}, ..., N'_{2m}$ w.r.t. D and a j such that $k < j \leq m$ and N'_{2j} is a-consistent w.r.t. D.*

It is also useful to define a *stable state* as a state that does not always immediately cause its own termination (note that stable states can be in loops, but must be a-consistent):

Definition 15 (Stable State). *Let D be a domain description and let S be a state. S is* ***stable w.r.t.*** *D if there exists a node $\langle S, \emptyset, P\rangle$ which is a proper causal descendant of $\langle S, \emptyset, \emptyset\rangle$.*

Example 8 (Promotion). An employee gets promoted at time 1. Promotion results in a large office (LO) and big salary (BS). But nobody gets a large office when the building is overcrowded (OC), which it is at time 1:

$\texttt{always}\ \neg(OC \wedge LO)$ (PR1)
$Promote\ \texttt{causes}\ \{BS, LO\}$ (PR2)
$Promote\ \texttt{occurs-at}\ 1$ (PR3)
$(\neg LO \wedge \neg BS \wedge OC)\ \texttt{holds-at}\ 1$ (PR4)

Here is the tree of possible causal chains that arise at time 1 in this example, with the single proper causal descendant of the root node underlined:

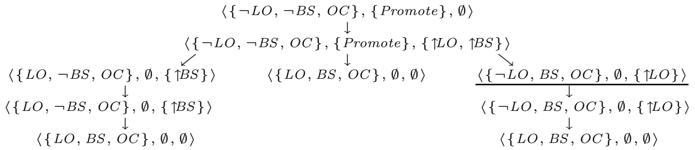
$\langle\{\neg LO, \neg BS, OC\}, \{Promote\}, \emptyset\rangle$
$\langle\{\neg LO, \neg BS, OC\}, \{Promote\}, \{\Uparrow LO, \Uparrow BS\}\rangle$
$\langle\{LO, \neg BS, OC\}, \emptyset, \{\Uparrow BS\}\rangle$
$\langle\{LO, BS, OC\}, \emptyset, \emptyset\rangle$
$\underline{\langle\{\neg LO, BS, OC\}, \emptyset, \{\Uparrow LO\}\rangle}$
$\langle\{LO, \neg BS, OC\}, \emptyset, \{\Uparrow BS\}\rangle$
$\langle\{\neg LO, BS, OC\}, \emptyset, \{\Uparrow LO\}\rangle$
$\langle\{LO, BS, OC\}, \emptyset, \emptyset\rangle$
$\langle\{LO, BS, OC\}, \emptyset, \emptyset\rangle$

Fig. 2.

3.2 Definitions Regarding Time and Temporal Change

If a causal node corresponds to a particular time-point in the narrative of a given domain description (e.g. in Fig. 1, N_0 corresponds to time 2), then Definitions 16 – 21 below ensure that the states within its proper causal descendants indicate possible choices as to which fluents will change values in the time period immediately afterwards. These definitions are largely modifications of those in [7], but with the notion of a *change set* replacing that of initiation/termination points.

Definition 16 (Interpretation). *An* ***interpretation*** *of $\mathcal{ME}$ is a mapping $H : \Phi \times \Pi \mapsto \{true, false\}$.*

Definition 17 (Time-point Satisfaction). *Given a fluent formula ϕ of $\mathcal{ME}$ and a time point T, an interpretation H **satisfies** ϕ **at** T iff the mapping M_T defined by $\forall F, M_T(F) = H(F,T)$ is a model of ϕ. Given a set Z of fluent formulae, H **satisfies** Z **at** T iff H satisfies ϕ at T for each $\phi \in Z$.*

Definition 18 (State/Event Base at a Time-point). *Let D be a domain description, H an interpretation, and T a time-point. The **state at** T **w.r.t.** H, denoted $S(H,T)$, is the state $\{F \mid H(F,T) = true\} \cup \{\neg F \mid H(F,T) = false\}$. The **event base at** T **w.r.t.** D, denoted $B(D,T)$, is the set $\{A \mid$ "A `occurs-at` T" $\in D\}$.*

Definition 19 (Causal Frontier). *Let D be a domain description, T a time-point, H an interpretation and S a state. S is a **causal frontier of** H **at** T **w.r.t.** D iff there exists a node $N = \langle S, B, P \rangle$ such that N is a proper causal descendant of $\langle S(H,T), B(D,T), \emptyset \rangle$ w.r.t. D.*

Definition 20 (Change Set). *Let D be a domain description, H an interpretation, T a time-point and $\mathcal{C}$ a set of fluent literals. $\mathcal{C}$ is a **change set at** T **w.r.t.** H iff there exists a causal frontier S of H at T w.r.t. D such that $\mathcal{C} = S - S(H,T)$.*

Definition 21 (Model). *Let D be a domain description, and let Φ^* be the set of all (+ve and -ve) fluent literals in the language. Then an interpretation H is a **model** of D iff there exists a mapping $c : \Pi \mapsto 2^{\Phi^*}$ such that for all T, $c(T)$ is a change set at T w.r.t. H, and the following three conditions hold. For every fluent literal L and time-points $T_1 \prec T_3$:*

1. *If H satisfies L at T_1, and there is no time-point T_2 s.t. $T_1 \preceq T_2 \prec T_3$ and $\overline{L} \in c(T_2)$, then H satisfies L at T_3.*
2. *If $L \in c(T_1)$, and there is no time-point T_2 s.t. $T_1 \prec T_2 \prec T_3$ and $\overline{L} \in c(T_2)$, then H satisfies L at T_3.*
3. *H satisfies the following constraints:*
   - *For all "ϕ `holds-at` T" in D, H satisfies ϕ at T.*
   - *For all time-points T, $S(H,T)$ is a stable state.*

Intuitively, condition (1) above states that fluents change their truth values only via successful effects of c-propositions, and (2) states that successfully initiating a literal establishes its truth value as true. Note also that condition (3)'s requirement of stability ensures that $S(H,T)$ is a-consistent.

Definition 22 (Consistency and Entailment). *A domain description D is **consistent** if it has a model. D **entails** the h-proposition "ϕ `holds-at` T", written $D \models \phi$ `holds-at` T, iff for every model M of D, M satisfies ϕ at T.*

Example 9 (Faulty Circuit). An electric current in a faulty circuit is switched on causing a broken fuse, which in turn terminates the current:

$\{SwitchOn\}$ `causes` $ElectricCurrent$ (FC1)

$\{ElectricCurrent\}$ `causes` $BrokenFuse$ (FC2)

$\{\mathit{BrokenFuse}\}$ `causes` $\neg\mathit{ElectricCurrent}$ (FC3)
`always` $\neg(\mathit{ElectricCurrent} \wedge \mathit{BrokenFuse})$ (FC4)
$\mathit{SwitchOn}$ `occurs-at` 1 (FC5)

One causal chain that could be triggered at time 1 (with non-a-consistent nodes N_4 and N_5) is:

$N_0 : \langle\{\neg\mathit{ElectricCurrent}, \neg\mathit{BrokenFuse}\}, \{\mathit{SwitchOn}\}, \emptyset\rangle$
$N_1 : \langle\{\neg\mathit{ElectricCurrent}, \neg\mathit{BrokenFuse}\}, \{\mathit{SwitchOn}\}, \{\uparrow\mathit{ElectricCurrent}\}\rangle$
$N_2 : \langle\{\mathit{ElectricCurrent}, \neg\mathit{BrokenFuse}\}, \emptyset, \emptyset\rangle$
$N_3 : \langle\{\mathit{ElectricCurrent}, \neg\mathit{BrokenFuse}\}, \emptyset, \{\uparrow\mathit{BrokenFuse}\}\rangle,$
$N_4 : \langle\{\mathit{ElectricCurrent}, \mathit{BrokenFuse}\}, \emptyset, \emptyset\rangle$
$N_5 : \langle\{\mathit{ElectricCurrent}, \mathit{BrokenFuse}\}, \emptyset, \{\downarrow\mathit{ElectricCurrent}\}\rangle$
$N_6 : \langle\{\neg\mathit{ElectricCurrent}, \mathit{BrokenFuse}\}, \emptyset, \emptyset\rangle.$

This chain is well-formed because N_6 is the first static resolvant node and is fully resolved (Definition 13). N_6 is a-consistent and therefore is a proper causal descendant of N_0 (Definition 14). So $\{\neg\mathit{ElectricCurrent}, \mathit{BrokenFuse}\}$ is a causal frontier at 1 of any interpretation that satisfies $(\neg\mathit{ElectricCurrent} \wedge \neg\mathit{BrokenFuse})$ at 1 (Definition 19), thus providing the change set $\{\mathit{BrokenFuse}\}$ (Definition 20). Note that at the granularity level of the representation of this example, *ElectricCurrent*, the cause of *BrokenFuse*, is never true! *ElectricCurrent* is true only at a finer granularity.

4 Some Formal Results and Properties

As we have seen, $\mathcal{ME}$ provides principled, general mechanisms for causal laws to be qualified both by each other and by static laws, thus integrating all endogenous qualifications within one base-level semantic framework. $\mathcal{ME}$ also provides a high degree of modularity by its separation of information about causality (c-, p- and a-propositions), narrative information about attempted actions (o-propositions), and observations (h-propositions) within the narrative. These qualities make $\mathcal{ME}$ domain descriptions particularly elaboration tolerant, as the following results show. (Proofs of all results at www.ucl.ac.uk/slais/rob-miller/modular-e/lpnmr05long.pdf).

Definition 23 (Pre- and Post-observation/action Points). *Given a domain description D, a **post-observation point** of D is a time-point T_p such that, for every h-proposition of the form "ϕ `holds-at` T" in D, $T \preceq T_p$. A **pre-action point** (respectively **post-action point**) is a time-point T_a such that, for every o-proposition "A `occurs-at` T" in D, $T_a \preceq T$ (respectively $T_a \succeq T$).*

Definition 24 (Projection Domain Description). *The domain description D is a **projection domain description** if there exists a time-point which is both a post-observation point and a pre-action point of D.*

Theorem 1 (Free Will Theorem). *Let M be a model of a finite domain description D, let O be a finite set of o-propositions, and let T_n be a time-point which is both a post-observation point for D and a pre-action point for O. Then there is a model M_O of $D \cup O$ such that for any fluent F and time-point $T \preceq T_n$, $M_O(F,T) = M(F,T)$.*

Corollary 1 (Free Will Corollary). *Let D and D' be domain descriptions and let T_n be a post-observation point for both D and D'. Let D and D' differ only by o-propositions referring to time-points greater than or equal to T_n and let M be a model of D. The there is a model M' of D' such that $M(F,T) = M'(F,T)$ for all fluent constants F and all time-points T such that $T \preceq T_n$.*

Corollary 2 (Action Elaboration Tolerance Corollary). *Let D be a consistent domain description and let O be a finite set of o-propositions. If there exists a time-point T_n which is both a post-observation point for D and a pre-action point for O, then $D \cup O$ is consistent.*

Theorem 2 demonstrates the robustness and elaboration tolerance of $\mathcal{ME}$ theories by showing that their consistency is contingent only on the internal consistency of the static laws and on whether observations match with predicted effects.

Theorem 2 (Theorem of Causal Elaboration Tolerance). *Let D_a be a consistent domain description consisting only of a-propositions and let E be a finite set of o-, c- and p-propositions. Then $D_a \cup E$ is also a consistent domain description.*

Lack of space prevents us from giving further formal results here on the link between global and local qualifications as illustrated in examples 3 and 4. These results show their complex relationship and hence the difficulty to have modularity when a framework relies overly on explicit local qualification.

5 Exogenous Qualifications

$\mathcal{ME}$'s base semantics offers an elaboration tolerant solution to the endogenous qualification problem, where properties of the domain implicitly qualify the effect laws. It is, nonetheless, still possible that an effect fails to be produced as expected. Such a scenario occurs, for instance, when we elaborate Example 2 by observing the car not running at time 2. No known reason can explain this unexpected observation, so it needs to be attributed to an exogenous cause.

A way to reconcile such conflicts is to assume that every effect law of a domain description is implicitly qualified [4] by a set of extra preconditions, written *Normal_exo* that symbolizes the normal conditions under which the law operates successfully. These preconditions are outside the user's language or *exogenous* [13], and package together all the unknown conditions necessary for the effect law to successfully generate its effect. They hold true by default unless the observations in a given narrative make the domain description inconsistent.

Definition 25 (Default Domain Description). *Let D be a domain description. To obtain the* **default domain description** D_d **associated with** *D: (i) replace every c-proposition "C* causes *L" with "$C \cup Normal\_exo(C, L)$* causes *L", and (ii) add the* ***n-proposition*** *"*normally *$norm\_exo(\_)$" for every fluent $norm\_exo(\_)$ in some set $Normal\_exo(\_)$.*

The exogenous fluents *norm_exo* that belong to the *Normal_exo* sets depend on assumptions on the nature of the failures of the effect law, in the particular domain of application. A meta-level **recovery policy** can be chosen a-priori appropriate for the domain at hand. Omitting the details, a recovery policy defines what other effect laws will be assumed to fail once a given effect law is observed to fail. One can define recovery policies where (i) no other effect laws are assumed to fail, (ii) all effect laws sharing the same effect L are also assumed to fail, (iii) all effect laws sharing the same event set C are also assumed to fail, etc. Irrespective of the recovery policy, the default models of domain D are given via the same simple minimization of the exogenous fluents over the (strict) models of the associated default domain D_d.

Definition 26 (Default Model). *Let D be a domain description, T_a a pre-action point of D, and D'_d the default domain description associated with D but without its n-propositions. Then, the restriction of M to fluents other than the $norm\_exo(\_)$ fluents is a* **default model** *of D iff (1))M is a model of D'_d, and (2) There is no model M' of D'_d such that $N \subset N'$, where*

$$N = \{norm\_exo(\_) \mid M(norm\_exo(\_), T_a) = true\},$$
$$N' = \{norm\_exo(\_) \mid M'(norm\_exo(\_), T_a) = true\}.$$

So far we have assumed that once an effect law is observed to fail, all subsequent instances will also fail by virtue of the persistence of *norm_exo* fluents. Various alternatives are also possible and the semantics can easily be adapted to support them. An observed failed effect law might, for example, cause its subsequent instances to fail nondeterministically, or not fail at all. Hence, in addition to failures, we can also have uncertain failures, or "accidents" (see [13]).

The existence of default models can be guaranteed as long as domains are a-consistent, point-wise consistent w.r.t. h-propositions, and do not violate fluent persistence. This requirement is captured by the notion of a *frame model*, which (assuming a "coupled accidents" recovery policy, where the exogenous qualification of a causal law exactly implies the exogenous qualification of all other causal laws applied to the same time-point) can be defined similarly to a model with the exception that the change set mapping $c(\cdot)$ can map arbitrary time-points to the empty set. Intuitively, this frame model definition allows all causal laws at some time-point to simply fail to produce their effects, as long as the successful production of their effects is not required to explain the change in the truth-value of some fluents.

Theorem 3 (Default Model Existence). *A domain description D has a default model iff it has a frame model.*

6 Summary and Related and Future Work

We have shown how $\mathcal{ME}$ can represent non-deterministic narrative domains involving concurrency, static laws and indirect effects. We have formally characterized $\mathcal{ME}$'s high degree of modularity and elaboration tolerance, enabled by an exceptionally full solution to the ramification problem able to deal with looping systems of indirect effects, and race conditions between competing causal laws. These properties help separate out, and provide a principled solutions to, the endogenous and exogenous qualification problems. Endogenous qualifications may be either locally specified or globally derived within the base semantics, whereas exogenous qualifications are provided by the use of default minimization.

Our approach to the qualification problem and its links to ramifications partly follows that in [13]. But $\mathcal{ME}$'s fuller solution to the frame problem, which covers both successful and failed action attempts, enables it to use the same default reasoning mechanism to deal with not just the "weak" but also the "strong" qualification problem as described in [13]. Two other important aspects in which $\mathcal{ME}$ differs from [13] are (a) the more complete treatment of ramifications, e.g. for concurrent effects and (b) the notion of global qualification which gives $\mathcal{ME}$ a higher degree of modularity. Our results are in line with the recent study of modularity in [6] which again highlights the link between modularity and free-will properties.

Our solution to the ramification problem is related to that in [1] in that the indirect effects of actions are defined constructively through causal laws. But $\mathcal{ME}$'s processed-based semantics differs in that it (a) embraces nondeterminism resulting from the possible orderings by which effects are realized, and (b) attributes meaning to domains (e.g, Example 9) that are deemed ill-formed in [1].

Irrespective of the qualification problem, the "free will" property of Theorem 1 is important to avoid *anomalous planning*, whereby unintended "plans" can be abduced or deduced for the converse of a precondition of an effect law by virtue of a lack of model for the successful application of that law. (See [11] for an example.) Although lack of space prevents an illustration here, anomalous plans are easy to construct in formalisms such as the Language C [5] which express action non-executability in terms of inconsistency. But they also arise in any framework (such as [7]) unable to provide models for all combinations of causal laws.

We currently have a prototype implementation of $\mathcal{ME}$'s base semantics in Prolog. The declarative programming style should facilitate an easy proof of the soundness and completeness of the implementation w.r.t. to $\mathcal{ME}$'s semantics. On the other hand, different techniques might be needed to address the *computational qualification problem* [3] of avoiding considering the majority of qualifications during the computation. Similar techniques might also prove useful when computing default models, where one does not want to consider all possible ways causal laws might fail, but rather deduce which ones should fail. To this end we are currently considering the use of satisfiability methods or Answer Set Programming (along the lines of [8,12,2]), as well as argumentation (or abduction) based computational methods. We also aim to study subclasses (as in [2]) of $\mathcal{ME}$, where the computational complexity of reasoning decreases.

There are several aspects of $\mathcal{ME}$ that deserve further study. One is the extent to which static laws should be regarded as specific to the temporal granularity of the representation (how would we refine the role that a-propositions play in computing indirect effects?). A detailed comparison would also be useful on different recovery policies used in $\mathcal{ME}$'s approach to the exogenous qualification problem. We would also like to investigate the use of priority policies on different $\mathcal{ME}$ models, e.g. to prefer non-change in nondeterministic situations.

Acknowledgement. This work was partially supported by the IST programme of the EC, FET under the IST- 2001-32530 SOCS project, within the Global Computing proactive initiative.

References

1. M. Denecker, D. Theseider-Dupré, and K. Van Belleghem. An inductive definition approach to ramifications. *Linkoping Electronic Articles in Computer and Information Science*, 3(7):1–43, 1998.
2. Y. Dimopoulos, A. Kakas, and L. Michael. Reasoning about actions and change in answer set programming. In *Proceedings of LPNMR'03*, 2004.
3. C. Elkan. On solving the qualification problem. In *Extending Theories of Actions: Formal theory and practical applications, AAAI Spring Symposium*, 1995.
4. M. Ginsberg and D. Smith. Reasoning about Action II: The Qualification Problem. *AIJ*, 35, 1988.
5. E. Giunchiglia, J. Lee, V. Lifschitz, N. McCain, and H. Turner. Nonmonotonic Causal Theories. *AIJ*, 153(1–2):49–104, 2004.
6. A. Herzig and I. Varzinczak. Domain Descriptions Should be Modular. In *Proceedings of ECAI 2004*. IOS Press, 2004.
7. A. Kakas and R. Miller. Reasoning about Actions, Narratives and Ramifications. *Journal of Electronic Transactions on AI*, 1(4), 1998.
8. H. Kautz and B. Selman. Pushing the envelope: Planning, propositional logic, and stochastic search. In *Proceedings of NCAI'96*, 1996.
9. V. Lifschitz. Missionaries and cannibals in the causal calculator. In *Proceedings of KR-2000*, 2000.
10. J. McCarthy. Elaboration tolerance. `http://www-formal.stanford.edy/jmc/elaboration` , 1999.
11. R. Miller and M. Shanahan. Some Alternative Formulations of the Event Calculus. *Lecture Notes in Artificial Intelligence*, 2408:452–490, 2002.
12. M. Shanahan and M. Witkowski. Event Calculus Planning Through Satisfiability. *J. of Logic Comp.*, 14(5):731–745, 2004.
13. M. Thielscher. The Qualification Problem: A solution to the problem of anomalous models. *AIJ*, 131(1–2):1–37, 2001.