Temporal conceptual models — preferences for representations

C. Maria Keet and Sonia Berman

Department of Computer Science, University of Cape Town,
South Africa, {mkeet,sonia}@cs.uct.ac.za

1 Introduction

These introductory notes describe the task, a refresher on the notation of the semantics of the constraints, and some explanatory notes on how to read the sample sentences and diagrams.

1.1 Task

For each element or constraint, fill in the corresponding section in the spreadsheet. The fields that require a value are colour-coded in red. You may either select from the drop-down list in column E or type a value between 0 and 5, where "5" is the most preferred, "1" least preferred, and "0" not at all. Any comments you may have can be added in column G.

1.2 Refresher notes on logic notation of the semantics

We rely on the formal foundations of ER_{VT} and EER^{++}_{VT} (with some adjustments to harmonise notation and the extension with mandatory constraints), being the semantics of $\mathcal{DLR}_{\mathcal{US}}$, so as not to clutter this presentation with too much repetition and syntax notation. Considering usual model-theoretic semantics, we use here a *temporal interpretation* of the signature of a conceptual data model $\mathcal{M}$ for this. This is a structure of the form: $\mathcal{I} = \big((\mathbb{Z}, <), \Delta^{\mathcal{I}}, \{\cdot^{\mathcal{I}(t)} \mid t \in \mathbb{Z}\}\big)$, where $(\mathbb{Z}, <)$ is the set of integers denoting the intended *flow of time*, $\Delta^{\mathcal{I}} \neq \emptyset$ is the *interpretation domain* divided into $\Delta^{\mathcal{I}}_C$ over classes and $\Delta^{\mathcal{I}}_D$ over data types, and $\cdot^{\mathcal{I}(t)}$, for $t \in \mathbb{Z}$, is the *interpretation function* which assigns a set $C^{\mathcal{I}(t)} \subseteq \Delta^{\mathcal{I}}$ to each entity type $C \in \mathcal{C}$, a set $R^{\mathcal{I}(t)}$ of tuples over $\Delta^{\mathcal{I}}_C \times \Delta^{\mathcal{I}}_C$ to each relation $R \in \mathcal{R}$ and a set $A^{\mathcal{I}(t)}$ of tuples over $\Delta^{\mathcal{I}}_C \times \Delta^{\mathcal{I}}_D$ to each attribute $A \in \mathcal{A}$. While $\mathcal{DLR}_{\mathcal{US}}$ permits n-ary relations, we consider here just the case for binaries.

So, for instance, a semantics like $o \in C^{\mathcal{I}(t)} \rightarrow \forall t' \in \mathcal{T}.o \in C^{\mathcal{I}(t')}$ states that an object o is a member of the temporal interpretation (the "$\mathcal{I}(t)$") of class C at time t, and if that holds, then (the "$\rightarrow$") for all times t' in the set of time point $\mathcal{T}$, object o is still a member of class C; i..e, it holds at all time points in the past, present, and in the future. Another example, for relationships, is $r \in R^{\mathcal{I}(t)} \rightarrow \exists t' \neq t.r \notin R^{\mathcal{I}(t')}$, where r is a binary relation (i.e., with two participating objects, member of a class) instantiating the relationship $\mathcal{R}$ at time t, and if that holds, then there is a time point t' in the set of time point $\mathcal{T}$ that is not time point t, where r is *not* a member of the relationship $\mathcal{R}$; i.e., there's some time in the past or in the future where it doesn't hold.

Regarding $\mathcal{DLR}_{\mathcal{US}}$ notation, that is just like the description logics notation you have seen before in OE and/or LAI, but then with temporal operators. The temporal counterpart to

$\exists$ is $\Diamond^*$ and the temporal counterpart to the $\forall$ is $\Box^*$. Added to that is a $+$ or a $-$ for the future and the past, respectively. So, then a $\Diamond^+$ means 'some time in the future this will be true' and a $\Diamond^-$ means 'this was true some time in the past', and a $\Diamond^{-3}$ 'this was true three time units ago', where a time unit could be a year, week, etc. Further, there's a notion of 'the (immediate) next time instant', as $\oplus$, and 'the (immediate) previous time instant', as $\ominus$. For instance, we know that students are at some point in time not students (you weren't before going to school, and at some point you'll graduate and are thus not a student anymore), so that is represented in a temporal DL as $\mathsf{Student} \sqsubseteq \Diamond^* \neg \mathsf{Student}$. There are also transition constraints, which are written in a shorthand notation (syntactic sugar) as, e.g., $\textsc{Dex}_{\mathsf{Class1},\mathsf{Class2}}$, which in full DL notation is $\mathsf{Class1} \sqcap \neg \mathsf{Class2} \sqcap \oplus \mathsf{Class2}$, and $\textsc{Dev}_{\mathsf{Class1},\mathsf{Class2}}$ is the shorthand notation for $\mathsf{Class1} \sqcap \neg \mathsf{Class2} \sqcap \oplus (\neg \mathsf{Class1} \sqcap \mathsf{Class2})$.

1.3 Sample diagram

Besides the logic notation, there is also a graphical notation. This is basically a flavour of ER diagrams that has been extended with the temporal constraints. That notation has gone through some evaluation in previous research, and those participants preferred a 'clock' notation attached to an entity type (DL class), relationship (DL role), or attribute, and arrows with a dashed shaft for the transition constraints between entity types and between relationships if it is an optional transition, and a solid shaft for a mandatory transition. An example is shown in Figure 1 for illustrative purpose, noting that we are in the scope of conceptual data modelling, not ontologies, so the the constraints hold for a specific universe of discourse ('business rules' of a company) and intended for some particular database, rather than what generally must hold for a subject domain.

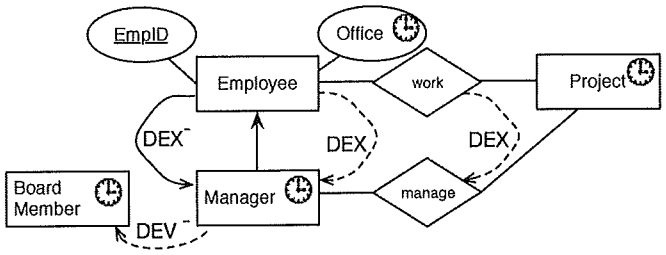


Fig. 1. Example of a temporally extended ER diagram in an updated EER^{++}_{VT} notation.

E.g. Office is a temporal attribute, for an employee may have an office sometimes and not have an office at other times. The mandatory transition $\textsc{Dex}^-$ indicates that a manager must have been working for the company as a regular employee before being promoted to manager, and thus that that transition from employee to manager happened in the past. Not all employees get promoted to manager, hence, the optional Dex from employee to manager. Likewise, the transition from work to manage is optional.

1.4 Explanatory notes on the sentences

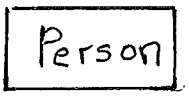
Each element/constraint has an abbreviation and a name. For most constraints, the semantics are given, with the others following the same pattern as the others of the same type. Several constraints also have an example from some hypothetical universe of discourse of a temporal conceptual model, to help filling in the slots of the template sentences. So, for instance in the first case, snapshot class (Sc), a sample entity type is Person, which would then be plugged in into the template, resulting in, e.g., "*Each* Person *is always a* Person.".

Regarding relationships, while the semantics doesn't constrain the classes (does not mention specific entity types), they are always considered in conceptual data models, mainly because there is a practice of reusing labels for naming of relationships (e.g., multiple times 'has') so that such context is necessary in the verbalisation of conceptual models (cf. ontologies). Therefore, we include the entity types in the sentences.

2 Elements and constraints for evaluation

2.1 Basics: snapshot and temporal

(Sc) Snapshot class; for instance, Person:
(a) $o \in Person^{\mathcal{I}(t)} \rightarrow \forall t' \in \mathcal{T}.o \in Person^{\mathcal{I}(t')}$
(b) Person $\sqsubseteq \Box^*$Person
(c) `Sc(Person)`
(d) Diagram:



(e) Person *is an entity type whose objects will always be a(n)* Person.

(Tc) Temporary class; for instance, Undergraduate Student (assuming they graduate and end up as alumni or drop outs):
(a) $o \in UndergraduateStudent^{\mathcal{I}(t)} \rightarrow \exists t' \neq t.o \notin UndergraduateStudent^{\mathcal{I}(t')}$
(b) UndergraduateStudent $\sqsubseteq \Diamond^* \neg$UndergraduateStudent
(c) `Tc(Undergraduate Student)`
(d) Diagram:

(e) *Each* Undergraduate Student *is not a(n)* Undergraduate Student *for some time.*

(SR) Snapshot relationship. For instance, Producer produces Movie.

(a) $r \in produces^{\mathcal{I}(t)} \rightarrow \forall t' \in \mathcal{T}.r \in produces^{\mathcal{I}(t')}$, where $(o, o') \in R$, and $o \in Producer$ and $o' \in Movie$ may be assumed to be declared.

(b) produces $\sqsubseteq \Box^*$produces (idem ditto on the producer and movie)

(c) `Sr(produces)` (idem ditto on the producer and movie)

(d) Diagram:

(e) *Each* Producer produces Movie *endures indefinitely.*

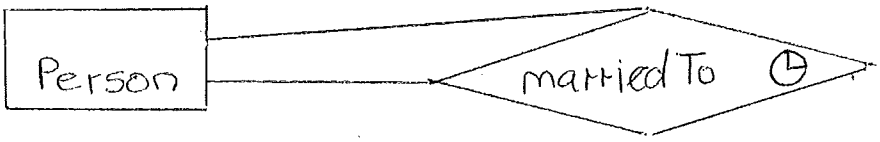
(TR) Temporal relationship. For instance, Person married-to Person in a conceptual model for a census database (as there is assumed to be either a divorce or widowhood at some point).

(a) $r \in marriedTo^{\mathcal{I}(t)} \rightarrow \exists t' \neq t.r \notin marriedTo^{\mathcal{I}(t')}$, where $(o, o') \in marrriedTo$, and $o \in Person$ and $o' \in Person$ may be assumed to be declared.

(b) marriedTo $\sqsubseteq \Diamond^* \neg$marriedTo (idem ditto on the persons participating in it)

(c) `Tr(marriedTo)` (idem ditto on the person participating in it)

(d) Diagram:



(e) *The objects participating in a fact in* Person married to Person *do* not *relate through* married-to *at some time.*

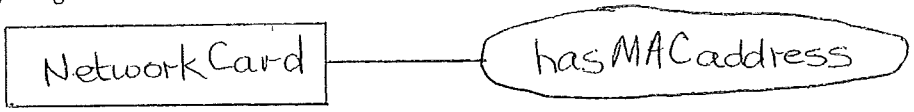
(SA) Snapshot attribute, For instance, Network Card has MAC Address `String`, where there attribute A_1 is has MAC Address. Note that the template would drop '*has*' if it is already in the attribute's name, or it is substituted with the verb in the attribute's name:

(a) $o \in NetworkCard^{\mathcal{I}(t)} \wedge \langle o, d \rangle \in hasMACAddress^{\mathcal{I}(t)} \rightarrow \forall t' \in \mathcal{T}.\langle o, d \rangle \in hasMACAddress^{\mathcal{I}(t')}$

(b) hasMACAddress $\sqsubseteq \Box^*$hasMACAddress

(c) `Sa(hasMACAddress)`

(d) Diagram:



(e) *Each object in entity type* Network Card *having attribute* has MAC Address *has* MAC Address *at all times.*

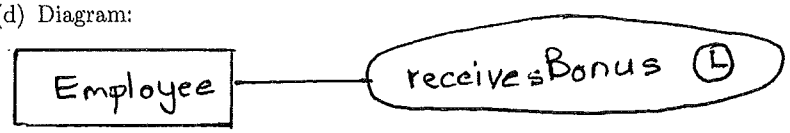
(TA) Temporal attribute. For instance, the temporal counterpart for this constraint with Employee receives Bonus Integer.

(a) $o \in Employee^{\mathcal{I}(t)} \wedge \langle o, d \rangle \in receivesBonus^{\mathcal{I}(t)} \rightarrow \exists t' \neq t. \langle o, d \rangle \notin receivesBonus^{\mathcal{I}(t')}$

(b) receivesBonus $\sqsubseteq \Diamond^* \neg$receivesBonus

(c) `Ta(receivesBonus)`

(d) Diagram:



(e) *Each object in entity type* Employee *having attribute* receives Bonus *does not have a(n)* receives Bonus *at some time.*

3 Dynamic constraints

3.1 Dynamic constraints for classes

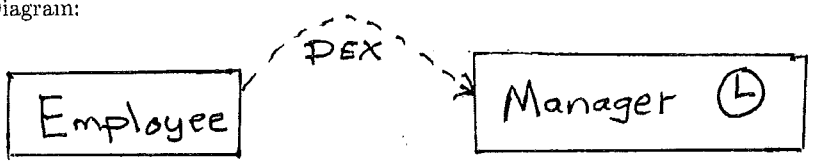
(DEX) Dynamic extension in the future, i.e.,

a. $o \in Employee^{\mathcal{I}(t)} \wedge o \notin Manager^{\mathcal{I}(t)} \wedge o \in Manager^{\mathcal{I}(t+1)}$

b. $\Diamond^+ \text{DEX}_{\text{Employee,Manager}}$

c. `Dex(Employee, Manager)`

d. Diagram:



e. *A(n)* Employee *may also become a(n)* Manager .

(DEXM) Mandatory DEX, i.e., apply, meaning that *all* objects instantiating C_1 will also instantiate C_2 at some point in time in the future: For instance, each PhDThesis will be added to the library as a LibraryBook.

a. $o \in PhDThesis^{\mathcal{I}(t)} \rightarrow \exists t' > t. o \in \text{DEX}^{\mathcal{I}(t')}_{PhDThesis, LibraryBook}$

b. PhDThesis $\sqsubseteq \Diamond^+ \text{DEX}_{\text{TPhDThesis,LibraryBook}}$

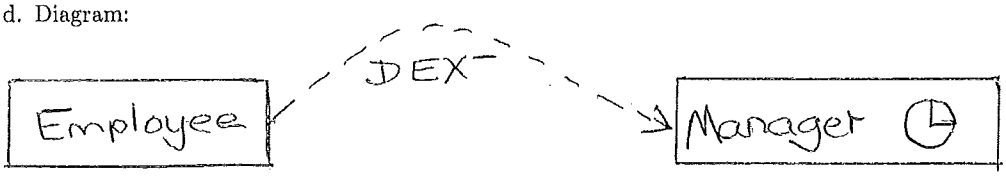
c. `DexM(PhDThesis, LibraryBook)`

d. Diagram:

e. *Each* PhDThesis *also has to become* a(n) LibraryBook .

(DEX$^-$) Dynamic extension in the past, i.e., For instance, a Manager at a company was already an Employee at that company:

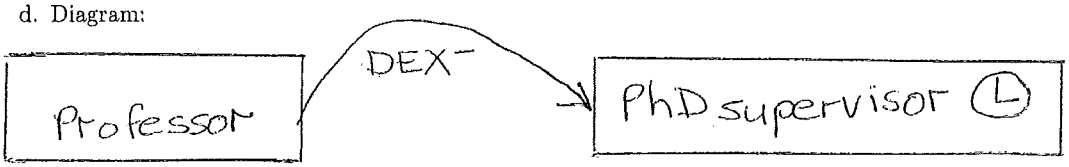
a. $o \in Employee^{\mathcal{I}(t)} \wedge o \in Manager^{\mathcal{I}(t)} \wedge o \notin Manager^{\mathcal{I}(t-1)}$
b. $\Diamond^- \text{DEX}_{\text{Employee,Manager}}$
c. `Dex⁻(Employee,Manager)`
d. Diagram:



e. *A(n)* Manager *may have been a(n)* Employee *before.*

(DEXM$^-$) Mandatory dynamic extension, past: For instance, one can only be a PhD supervisor if one is a Professor, and all professors are going to have to supervise PhD students, according to university management.

a. $o \in PhDsupervisor^{\mathcal{I}(t)} \rightarrow \exists t' < t.o \in \text{DEX}^{\mathcal{I}(t')}_{Professor,PhDsupervisor}$
b. $\text{PhDsupervisor} \sqsubseteq \Diamond^- \text{DEX}_{\text{Professor,PhDsupervisor}}$
c. `DexM⁻(Professor, PhD supervisor)`
d. Diagram:



e. *Each* PhD supervisor *was already a(n)* Professor .

(DEV) Dynamic evolution, future, optional: For instance, a Caterpillar is expected to turn into a Butterfly (just not all do in praxis).

a. $o \in Caterpillar^{\mathcal{I}(t)} \wedge o \notin Butterfly^{\mathcal{I}(t)} \wedge o \in Butterfly^{\mathcal{I}(t+1)} \wedge o \notin Caterpillar^{\mathcal{I}(t+1)}$
b. $\Diamond^+ \text{DEV}_{\text{Caterpillar,Butterfly}}$
c. `Dev(Caterpillar,Butterfly)`
d. Diagram:

e. *A(n)* Caterpillar *may evolve to become a(n)* Butterfly *ceasing to be a(n)* Caterpillar .

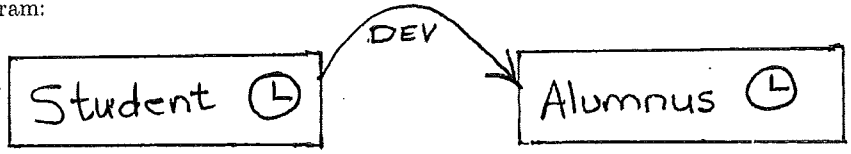
(DEVM) Mandatory dynamic evolution, future: For instance, at a university where each Student eventually will receive a certificate and thus become an Alumnus (and never study there again), the constraint for the database may be:

a. $o \in Student^{\mathcal{I}(t)} \rightarrow \exists t' > t.o \in \text{DEV}^{\mathcal{I}(t')}_{Student,Alumnus}$

b. $\text{Student} \sqsubseteq \Diamond^{+}\text{DEV}_{\text{Student,Alumnus}}$

c. `DevM(Student, Alumnus)`

d. Diagram:



e. *Each* Student *must evolve to* Alumnus *ceasing to be a(n)* Student .

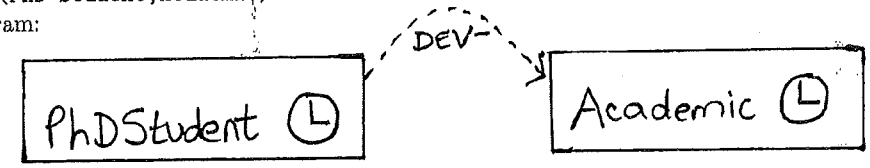
(DEV^{-}) Dynamic evolution, past, optional: For instance, for a university database: while it is common now for an Academic to move between universities, some stay at the same place, or return to, where they have been a PhD student.

a. $o \in Academic^{\mathcal{I}(t)} \wedge o \notin PhDStudent^{\mathcal{I}(t)} \wedge o \in PhDStudent^{\mathcal{I}(t-1)} \wedge o \notin Academic^{\mathcal{I}(t-1)}$

b. $\Diamond^{-}\text{DEV}_{\text{PhDStudent,Academic}}$

c. `Dev⁻(PhD Student,Academic)`

d. Diagram:



e. *A(n)* Academic *may have been a(n)* PhD Student *before, but is not a(n)* PhD Student *now.*

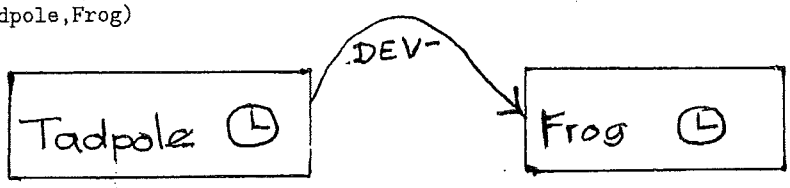
(DEVM^{-}) Mandatory dynamic evolution, past: For instance, Frog and the Tadpole it used to be.

a. $o \in Frog^{\mathcal{I}(t)} \rightarrow \exists t' < t.o \in \text{DEV}^{\mathcal{I}(t')}_{Tadpole,Frog}$

b. $\text{Frog} \sqsubseteq \Diamond^{-}\text{DEV}_{\text{Tadpole,Frog}}$

c. `DevM⁻(Tadpole,Frog)`

d. Diagram:



e. *Each* Frog *was a(n)* Tadpole *before, but is not a(n)* Tadpole *now.*

(PDEX/PDEV) Persistent extension or evolution; for instance, once a conference paper has been published in print, that Publication is there for posterity in, say, Springer's system. The persistence can come after any DEV/DEX variant introduced before, to which the following persistence constraint is added:

a. $o \in Publication^{\mathcal{I}(t)} \rightarrow \forall t' > t.o \in Publication^{\mathcal{I}(t')}$

b. Publication $\sqsubseteq \Box^{+}$Publication

c. `PDex(Publication)` or `PDev(Publication)`

d. Diagram:

e. <selected DEX/DEV option>, *and this remains so indefinitely.*

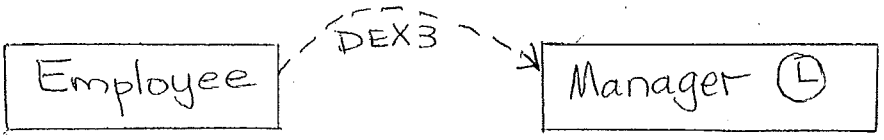
For quantitative extension and evolution, we need a specific number, n for counting and, implicitly, some time unit to be able to construct, e.g., 'after at least 3 years' and 'at most 5 months'. The number is denoted in the formalisation as n where $n \in \mathbb{Z}$ and $t + n \in \mathcal{T}_p$.
(QEX) Quantitative extension, future, optional, then: For instance, an Employee may be promoted to Manager at that company after 3 years of service.

a. $\exists(t+3) > t.(o \in Employee^{\mathcal{I}(t)} \wedge o \notin Manager^{\mathcal{I}(t)} \wedge Manager^{\mathcal{I}(t+3)})$

b. $\Diamond^{+3}\mathrm{QEX}_{\mathsf{Employee,Manager}}$

c. `Qex(Employee, Manager, 3 years)`

d. Diagram:



e. *A(n)* Employee *may also become a(n)* Manager *after* exactly 3 years .

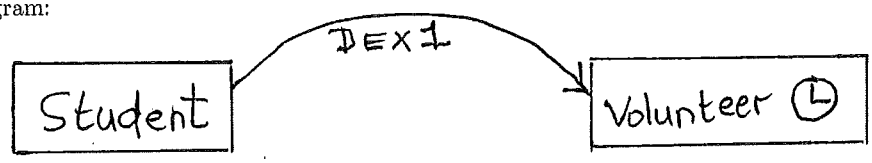
(QEXM) Quantitative extension, future, mandatory: For instance, all Students have to do some Volunteer service in the second year of their study.

a. $o \in Student^{\mathcal{I}(t)} \rightarrow \exists(t+1) > t.o \in \text{QEX}^{\mathcal{I}(t+1)}_{Student,Volunteer}$

b. $\text{Student} \sqsubseteq \Diamond^{+1}\text{QEX}_{\text{Student,Volunteer}}$

c. `QexM(Student, Volunteer, 1 year)`

d. Diagram:



e. *Each* Student *will also become a(n)* Volunteer *after* exactly 1 year .

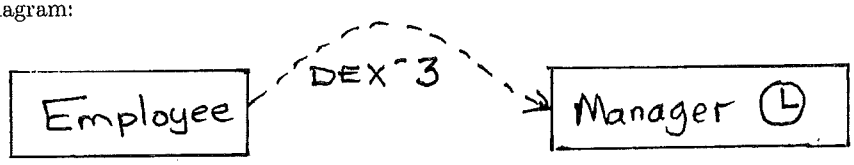
(QEX^{-}) Quantitative extension, past, optional, where the past counterpart is similar to the future, but then $t - n$.

a. $\exists(t-3) < t.(o \in Employee^{\mathcal{I}(t-3)} \wedge o \in Manager^{\mathcal{I}(t)} \wedge o \notin Manager^{\mathcal{I}(t-3)})$

b. $\Diamond^{-3}\text{QEX}_{\text{Employee,Manager}}$

c. `Qex⁻(Employee, Manager, 3 years)`

d. Diagram:



e. *A* manager *may already be a(n)* Employee *since* at least 3 years .

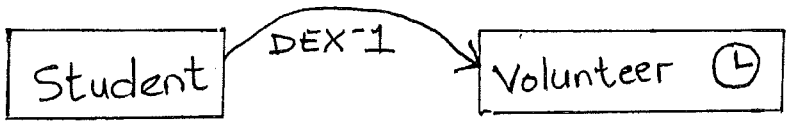
(QEXM^{-}) Quantitative extension, past, mandatory, where the past counterpart is similar to the future, but then $t' < t$:

a. $o \in Volunteer^{\mathcal{I}(t)} \rightarrow \exists(t-1) < t.o \in \text{QEX}^{\mathcal{I}(t-1)}_{Student,Volunteer}$

b. $\text{Volunteer} \sqsubseteq \Diamond^{-1}\text{QEX}_{\text{Student,Volunteer}}$

c. `QexM⁻(Student, Volunteer, 1 year)`

d. Diagram:



e. *Each* Volunteer *was already a(n)* Student *since* exactly 1 year .

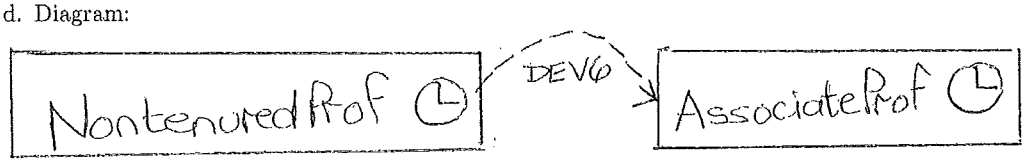
(QEV) Quantitative evolution, future: For instance, a non-tenured prof may have met the criteria and thus be confirmed as a Associate Prof after 6 years.

a. $\exists(t{+}6) > t.(o \in NonTenuredProf^{\mathcal{I}(t)} \wedge o \notin AssociateProf^{\mathcal{I}(t)} \wedge o \in AssociateProf^{\mathcal{I}(t+6)} \wedge o \notin NonTenuredProf^{\mathcal{I}(t+6)}$

b. $\Diamond^{+6}\text{QEV}_{\text{NonTenuredProf,AssociateProf}}$

c. `Qev(Non-tenured Prof, Associate Prof, 6 years)`

d. Diagram:



e. *A* Non-tenured Prof *may progress to a(n)* Associate Prof *after* exactly 6 years, *ceasing to be a(n)* Non-tenured Prof .

(QEVM) Quantitative evolution, future, mandatory): For instance, the fantasy world where all non-tenured profs will get that coveted job of associate prof after 6 years in the tenure track, regardless their performance.

a. $o \in NonTenuredProf^{\mathcal{I}(t)} \rightarrow \exists(t+6) > t.o \in \text{QEV}^{\mathcal{I}(t+6)}_{NonTenuredProf,AssociateProf}$

b. $\text{NonTenuredProf} \sqsubseteq \Diamond^{+6}\text{QEV}_{\text{NonTenuredProf,AssociateProf}}$

c. `QevM(Non-tenured Prof, Associate Prof, 6 years)`

d. Diagram:

e. *Each* Non-tenured Prof *must progress to a(n)* Associate Prof *after* exactly 6 years, *ceasing to be a(n)* Non-tenured Prof .

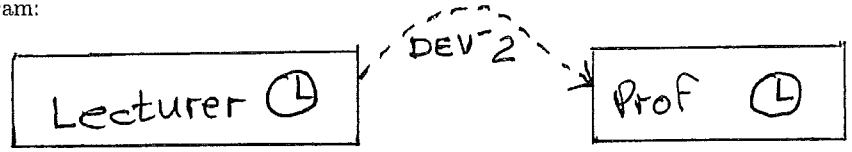
(QEV$^-$) Quantitative evolution, past, optional; e.g., a professor may have been a lecturer before for at least two years at the same institution

a. $\exists(t-2) < t.(o \in Prof^{\mathcal{I}(t)} \wedge o \notin Lecturer^{\mathcal{I}(t)} \wedge o \in Lecturer^{\mathcal{I}(t-2)} \wedge o \notin Prof^{\mathcal{I}(t-2)}$

b. $\Diamond^{-2}\text{QEV}_{\text{Lecturer,Prof}}$

c. `Qev⁻(Lecturer, Prof, 2 years)`

d. Diagram:



e. *A(n)* Prof *may have been a(n)* Lecturer *before for a period of* at least 2 years, *but is not now.*

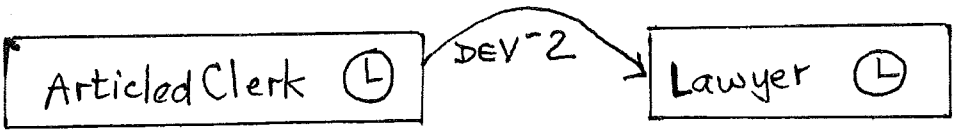
(QEVM$^-$) Quantitative evolution, past, mandatory; e.g., in an employee database of a law firm, each new lawyer has worked for two years as a articled clerk.

a. $o \in Lawyer^{\mathcal{I}(t)} \rightarrow \exists(t-2) < t.o \in \text{QEV}^{\mathcal{I}(t-2)}_{ArticledClerk,Lawyer}$

b. $\text{Lawyer} \sqsubseteq \Diamond^{-2}\text{QEV}_{\text{ArticledClerck,Lawyer}}$

c. `QevM⁻(Lawyer, ArticledClerk, 2 years)`

d. Diagram:



e. *Each* Lawyer *was a(n)* ArticledClerk *before for a period of* exactly 2 years, *but is not now.*

3.2 Dynamic constraints for relationships

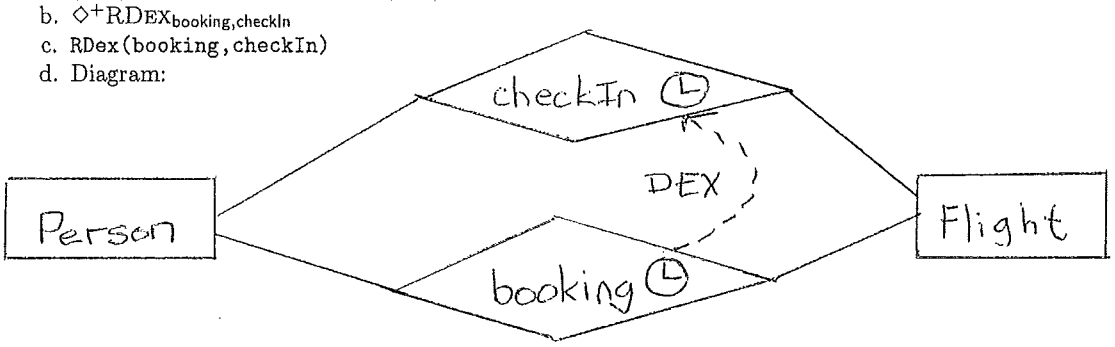
(RDEX) – Dynamic extension of a relationship. For instance, a Person has made a booking for a Flight, and then that person does a check-in for that flight some time afterward.

a. $\langle o, o' \rangle \in booking^{\mathcal{I}(t)} \rightarrow \exists t' > t. \langle o, o' \rangle \in checkin^{\mathcal{I}(t')}$

b. $\Diamond^{+}\text{RDEX}_{\text{booking,checkIn}}$

c. `RDex(booking,checkIn)`

d. Diagram:



e. Person booking Flight *may be followed by* Person check-in Flight *some time later.*

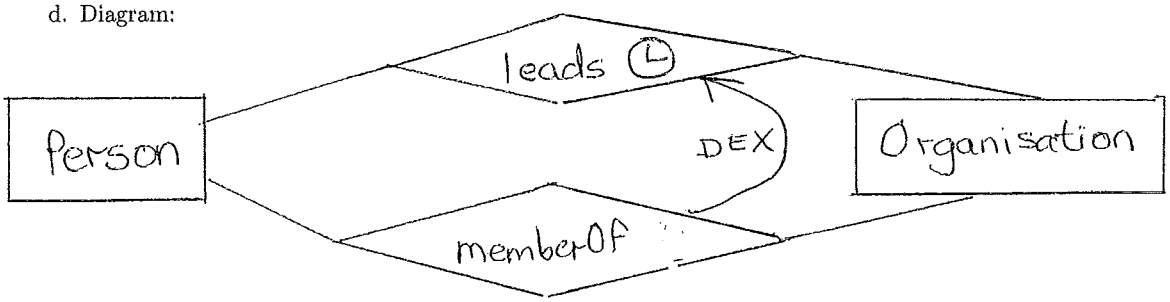
(RDEXM) – Dynamic extension for relationships, mandatory. For instance, there may be a rule in an organisation that everyone will take turns in leading the organisation.

a. $\langle o, o' \rangle \in memberOf^{\mathcal{I}(t)} \rightarrow \exists t' > t. \langle o, o' \rangle \in \text{RDEX}^{\mathcal{I}(t')}_{memberOf,leads}$

b. $\text{memberOf} \sqsubseteq \Diamond^{+}\text{RDEX}_{\text{memberOf,leads}}$

c. `RDex(member-of, leads)`

d. Diagram:



e. *Each* Person member-of Organisation *will be followed by* Person leads Organisation .

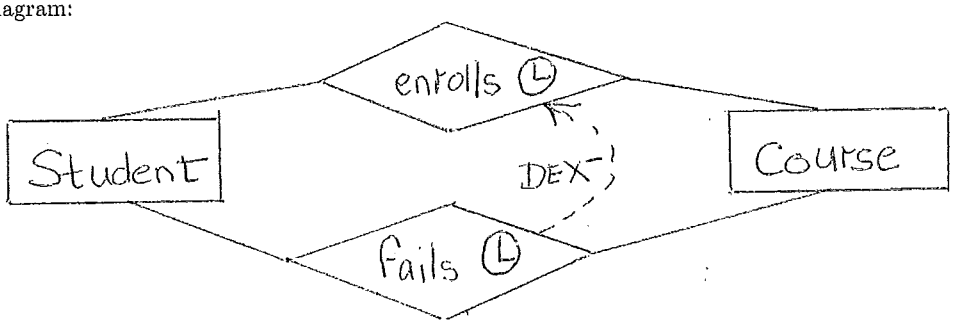
(RDEX$^-$) – Dynamic extension for relationships, past, optional. For instance, a student who failed a course may have enrolled in that course again (e.g., enrols in CSC1015F again [not considering the course as identified by 'CSC1015F in 2016']).

a. $\langle o, o' \rangle \in enrolls^{\mathcal{I}(t)} \rightarrow \exists t' < t . \langle o, o' \rangle \in fails^{\mathcal{I}(t')}$

b. $\Diamond^-\text{RDEX}_{\text{fails,enrolls}}$

c. `RDex⁻(fails,enrolls)`

d. Diagram:



e. Student enrolls in Course *may be preceded by* Student fails Course *some time earlier.*

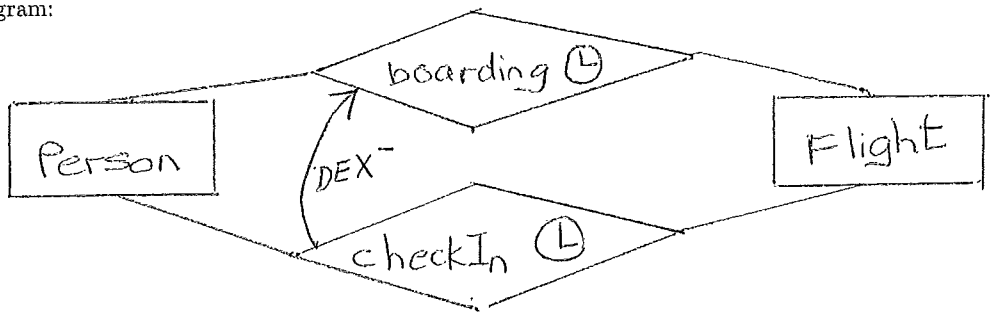
(RDEXM$^-$) – Dynamic extension for relationships, past, mandatory: For instance, every passenger who boards a flight must have checked in.

a. $\langle o, o' \rangle \in boarding^{\mathcal{I}(t)} \rightarrow \exists t' < t . \langle o, o' \rangle \in \text{RDEX}^{-\ \mathcal{I}(t')}_{checkIn,boarding}$

b. $\text{boarding} \sqsubseteq \Diamond^-\text{RDEX}_{\text{checkIn,boarding}}$

c. `RDex⁻(checkIn,boarding)`

d. Diagram:



e. *Each* Person boarding Flight *was preceded by* Person check-in Flight *some time earlier.*

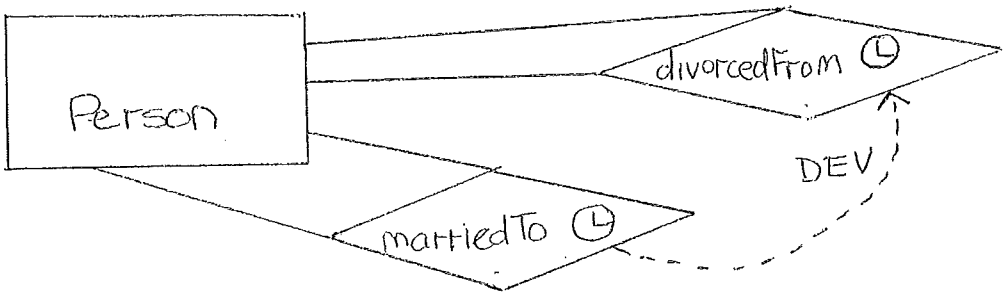
(RDEV) – Dynamic evolution for relationships, future, optional: For instance, a marriage between two persons may end up in a divorce of those two persons.

a. $\langle o, o' \rangle \in marriedTo^{\mathcal{I}(t)} \rightarrow \exists t' > t.\langle o, o' \rangle \in divorcedFrom^{\mathcal{I}(t')} \wedge \langle o, o' \rangle \notin marriedTo^{\mathcal{I}(t')}$

b. $\Diamond^{+}\text{RDEV}_{\text{marriedTo,divorcedFrom}}$

c. `RDev(married-to, divorced-from)`

d. Diagram:



e. Person married-to Person *may be followed by* Person divorced-from Person, *ending* Person married-to Person.

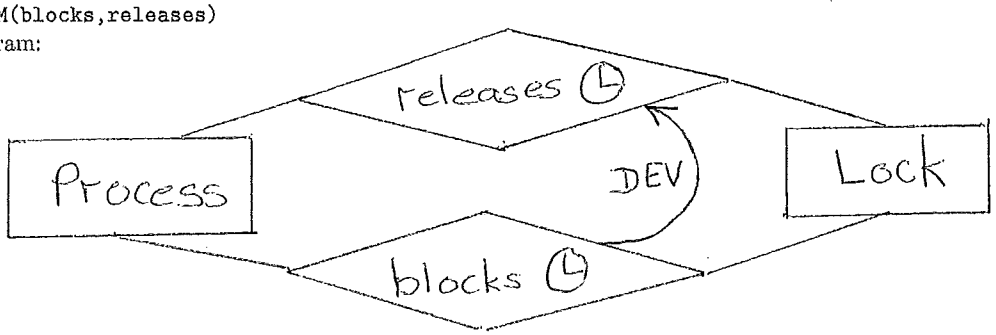
(RDEVM) – Dynamic evolution for relationships, future, mandatory: For instance, a process must release a lock after it has blocked that for a while.

a. $\langle o, o' \rangle \in \texttt{blocks}^{\mathcal{I}(t)} \rightarrow \exists t' > t.\langle o, o' \rangle \in \text{RDEV}^{\mathcal{I}(t')}_{blocks,releases}$

b. $\texttt{blocks} \sqsubseteq \Diamond^{+}\text{RDEV}_{\text{blocks,releases}}$

c. `RDevM(blocks,releases)`

d. Diagram:



e. *Each* Process blocks Lock *will be followed by* Process releases Lock , *terminating the* Process blocks Lock *relation.*

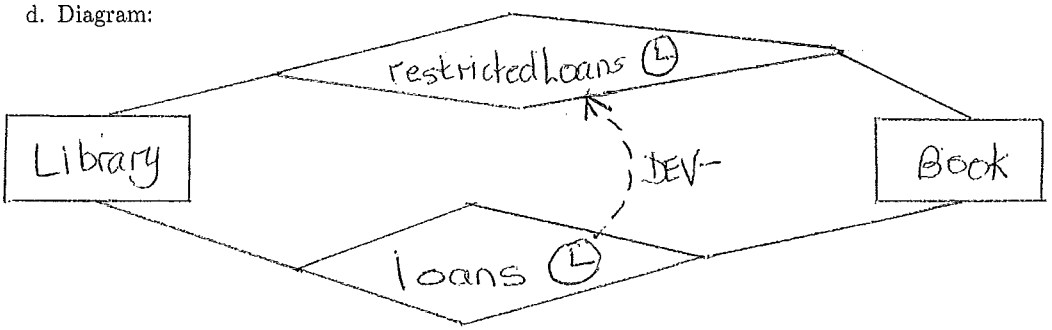
(RDEV$^-$) – Dynamic evolution for relationships, past, optional: For instance, books on restricted loan from a library may have been on regular loan from that library (e.g., a book is a high-in-demand textbook for a course in the semester, or it is from a special collection to begin with), and some couples remarry after they got divorced (like John Diggle and Laila in the TV series Arrow).

a. $\langle o, o' \rangle \in restrictedLoan^{\mathcal{I}(t)} \rightarrow \exists t' < t. \langle o, o' \rangle \in loan^{\mathcal{I}(t')} \wedge \langle o, o' \rangle \notin restrictedLoan^{\mathcal{I}(t')}$

b. $\Diamond^-\text{RDEV}_{\text{loan,restrictedLoan}}$

c. `RDev⁻(loan,restrictedLoan)`

d. Diagram:



e. Book on restricted loan from Library *may have been preceded by* Book on loan from Library *and they are not in that* Book on loan from Library *relation now.*

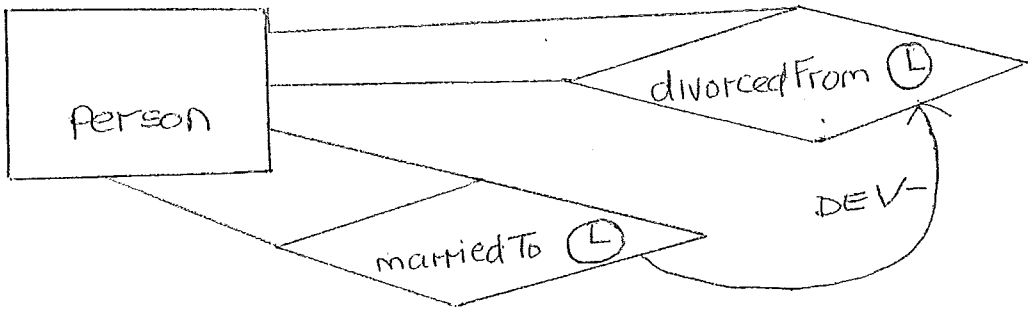
(RDEVM$^-$) – Dynamic evolution for relationships, past, mandatory: For instance, any pair of humans who are divorced were married before that.

a. $\langle o, o' \rangle \in dovircedFrom^{\mathcal{I}(t)} \rightarrow \exists t' < t. \langle o, o' \rangle \in \text{RDEV}^-_{married To, divorced From}{}^{\mathcal{I}(t')}$

b. $\text{divorcedFrom} \sqsubseteq \Diamond^-\text{RDEV}_{\text{marriedTo,divorcedFrom}}$

c. `RDevM⁻(marriedTo,divorcedFrom)`

d. Diagram:

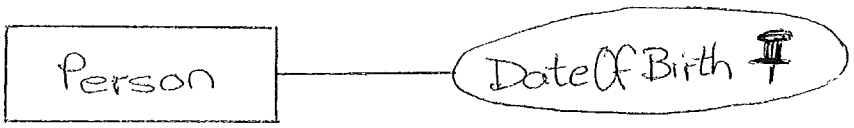


e. *Each* Person divorced-from Person *must have been preceded by* Person married-to person, *and terminating that* Person married-to Person *relation.*

(SRDEX/SRDev) – Persistence, optional or mandatory: same four options as for persistence with classes.

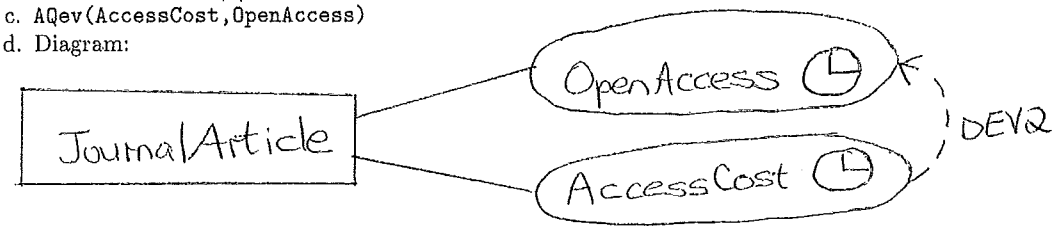
3.3 Dynamic constraints for attributes

(FREEZ) "frozen" attribute. For instance, the Date of Birth of a person is 'frozen' (persists) once that data is added to the database.
a. $\forall t' > t.a \in DateOfBirth^{\mathcal{I}(t')}$
b. DateOfBirth $\sqsubseteq \Box^*$DateOfBirth
c. `Freez(DateOfBirth)`
d. Diagram:



e. *Once the value for* DateOfBirth *is set, it cannot change anymore.*

(AQEV) Quantitative evolution, where a is a binary relation between a class and a data type, $a \in \text{AQEV}^{\mathcal{I}(t)}_{A_1,A_2} \rightarrow \exists(t+n) > t.(a \in \text{A}_1{}^{\mathcal{I}(t)} \wedge a \notin \text{A}_2{}^{\mathcal{I}(t)} \wedge a \in \text{A}_2{}^{\mathcal{I}(t+n)} \wedge a \notin \text{A}_1{}^{\mathcal{I}(t+n)})$ where $n \in \mathbb{Z}$. For instance, journal articles of some scientific journals have an 'embargo' of two years on them, where in the embargo period, one has to pay for reading the article, but afterward it becomes open access (accessible for free).
a. $\exists(t+2) > t.(a \in AccessCost^{\mathcal{I}(t)} \wedge a \notin OpenAccess^{\mathcal{I}(t)} \wedge a \in OpenAccess^{\mathcal{I}(t+2)} \wedge a \notin AccessCost^{\mathcal{I}(t+2)})$
b. $\Diamond^{+2}\text{AQEV}_{\text{AccessCost,OpenAccess}}$
c. `AQev(AccessCost,OpenAccess)`
d. Diagram:



e. *Each* JournalArticle's AccessCost *changes after* 2 years to OpenAccess.

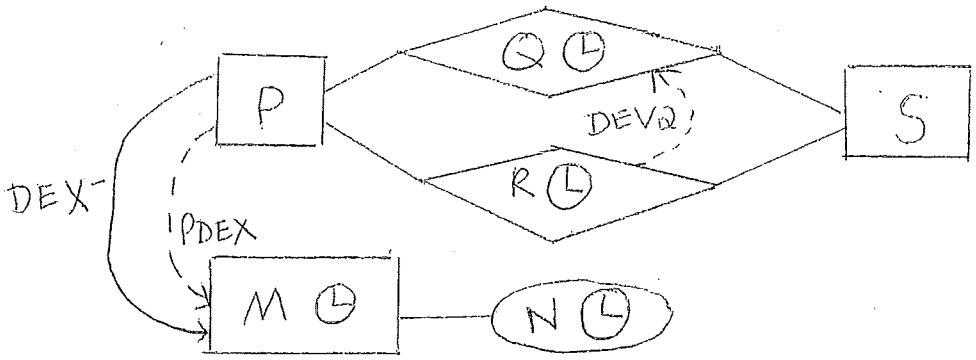
3.4 Interpreting the representation

For any, or all, of the following notations, describe in your own words what it means.

1. In the $\mathcal{DLR}_{\mathcal{US}}$ notation (A is an attribute; B, C, D are classes; Q, R are relationships):
   $\mathsf{A} \sqsubseteq \Diamond^{*}\neg\mathsf{A}$ $\mathsf{B} \sqsubseteq \Diamond^{*}\neg\mathsf{B}$
   $\mathsf{C} \sqsubseteq \Box^{*}\mathsf{C}$
   $\mathsf{D} \sqsubseteq \Box^{*}\mathsf{D}$
   $\mathsf{E} \sqsubseteq \Diamond^{*}\neg\mathsf{E}$
   $\mathsf{F} \sqsubseteq \Diamond^{*}\neg\mathsf{F}$
   $\mathsf{C} \sqsubseteq \Diamond^{+}\textsc{Dev}_{\mathsf{C,B}}$
   $\Diamond^{-}\textsc{Dev}_{\mathsf{C,B}}$
   $\Diamond^{+3}\textsc{Qex}_{\mathsf{Q,R}}$

2. The quasi-coding textual notation :
   Sc(I)
   Sc(L)
   Tc(G)
   Ta(G,H)
   Tr(J)
   Tr(K)
   `PDex(G)`
   `Qex`$^{-}$`(I, G, 2 years)`
   `RDevM(J,K)`

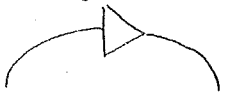
3. An EER^{++}_{VT} diagram:



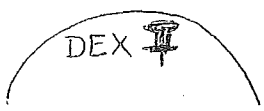
4 Diagram notation

While the notation for temporal elements and constraints used in the diagrams is a result of prior research, we would like to ask your opinion on the following additional possible notations that have been designed since.

1. General shape of the arrow for dynamic constraints:
   a. As used in the diagrams presented, with a plain solid or dashed shaft
   b. The shaft has a triangle in the middle, like this:



2. General shape for persistent extension or evolution:
   a. As used in the diagrams presented, where the arrow is labeled with PDEX or PDEV
   b. The arrow is labeled with the usual DEX or DEV and has a pin next to it, like this:



c. The arrow is labeled with the usual DEX or DEV and has a pin in the triangle, like this:

5 General questions

1. Is English your first/home language?
   [Yes / No]
2. Your level of education:
   Currently, I am a [Honours / Masters] student
3. Which course(s) are you currently doing or have you completed?
   [OE / LAI / both OE and LAI / neither]
4. Thank you for showing your notation preferences for understanding a given conceptual model. If you were designing a conceptual model yourself, rather than trying to understand someone else's model, which of the notations (a) to (e) would you prefer then?

5. Any other comments you may have: