Modeling the Evolution of Objects in Temporal Information Systems

Alessandro Artale

Faculty of Computer Science – Free University of Bolzano-Bozen

`artale@inf.unibz.it`

`http://www.inf.unibz.it/~artale/`

Joint work with:

Christine Parent–Université de Lausanne

Stefano Spaccapietra–EPFL Lausanne

LOA'05 – 14 July 2005

Introduction: Motivations

- Give a formalization based on set-theory of the various temporal constructs with particular attention to evolution constraints.
  - Clarify the meaning of the various temporal constructs;
  - Verify whether standard modeling requirements are verified;
  - Formal definition of quality criteria: Entity/Relationships/Schema consistency, Entity/Relationships Subsumption, Logical Implication;
  - Make explicit the implicit constraints in a model using the notion of logical implication.

Introduction: Modeling Requirements in a Temporal Setting

- Orthogonality. Temporal constructs should be specified separately and independently for classes, relationships, and attributes.
- Upward Compatibility. Preserve the non-temporal semantics of legacy conceptual schemas when embedded into temporal schemas.
- Snapshot Reducibility. A snapshot of the temporal database is described by the same schema without temporal constructs interpreted atemporally.
  - We should be able to fully rebuild a temporal database by starting from the single temporal snapshots.

Introduction: Temporal Conceptual Constructors

- Timestamping.

  The data model should distinguish between temporal and atemporal modeling constructs.

  – Realized by temporal marking of classes, relationships and attributes.

- Evolution Constraints.

  1. *Object Migration*: The possibility for an object to change its class membership;
  2. *Dynamic Relationships*: Either generate objects starting from other objects, or link objects existing at different times.

Outline

- Modeling Timestamping
- Modeling Evolution Constraints
  - Status Classes
  - Transitions
  - Generation Relationships
  - Cross-Time Relationships

$\mathcal{ER}_{VT}$: A Conceptual Model with Timestamps

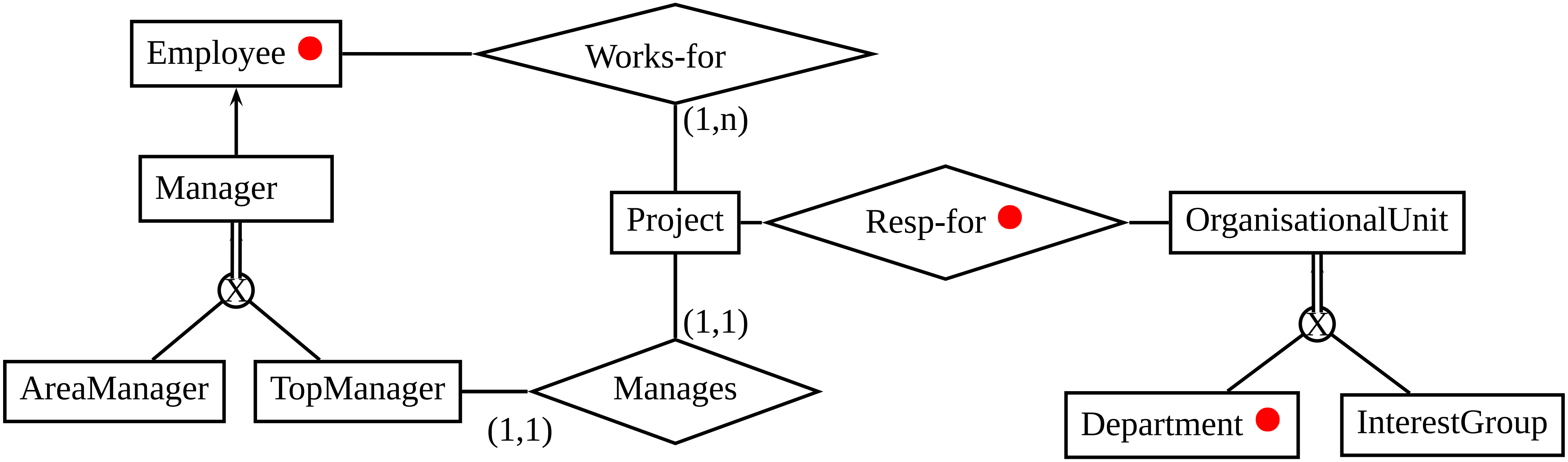
- $\mathcal{ER}_{VT}$ is equipped with both a linear and a graphical syntax along with a model-theoretic semantics.
- At the syntactical level, $\mathcal{ER}_{VT}$ supports timestamping of entities, relationships, and attributes using two different marks:
  - ***Snapshot*** constructs: Each of their instances have a global lifetime;
  - ***Temporary*** constructs: Each of their instances has a limited lifetime.

The Model-Theoretic Semantics for $\mathcal{ER}_{VT}$

A *temporal database state* for an $\mathcal{ER}_{VT}$ schema Σ is a tuple $\mathcal{B} = (\mathcal{T}, \Delta^{\mathcal{B}} \cup \Delta^{\mathcal{B}}_D, \cdot^{\mathcal{B}(t)})$:

- $\mathcal{T} = (\mathcal{T}_p, <)$, is the flow of time, where $\mathcal{T}_p$ is a set of time points (or chronons) and $<$ is a binary precedence relation on $\mathcal{T}_p$;
- $\Delta^{\mathcal{B}}$ is a nonempty set of abstract objects;
- $\Delta^{\mathcal{B}}_D$ is the set of basic domain values;
- $\cdot^{\mathcal{B}(t)}$ is a function that for each $t \in \mathcal{T}$ maps:
  - Every domain symbol D_i into a set $D_i^{\mathcal{B}(t)} = \Delta^{\mathcal{B}}_{D_i} \subseteq \Delta^{\mathcal{B}}_D$.
  - Every class C to a set $C^{\mathcal{B}(t)} \subseteq \Delta^{\mathcal{B}}$.
  - Every n-ary relationship R connecting the classes $C_1, \ldots, C_n$ to a set $R^{\mathcal{B}(t)}$, where $r \in R^{\mathcal{B}(t)} \rightarrow (r = \langle U_1 : o_1, \ldots, U_n : o_n \rangle \wedge \forall i \in \{1, \ldots, n\}. o_i \in C_i^{\mathcal{B}(t)})$.
  - Every attribute A to a set $A^{\mathcal{B}(t)} \subseteq \Delta^{\mathcal{B}} \times \Delta^{\mathcal{B}}_D$.

A Semantics for Timestamps



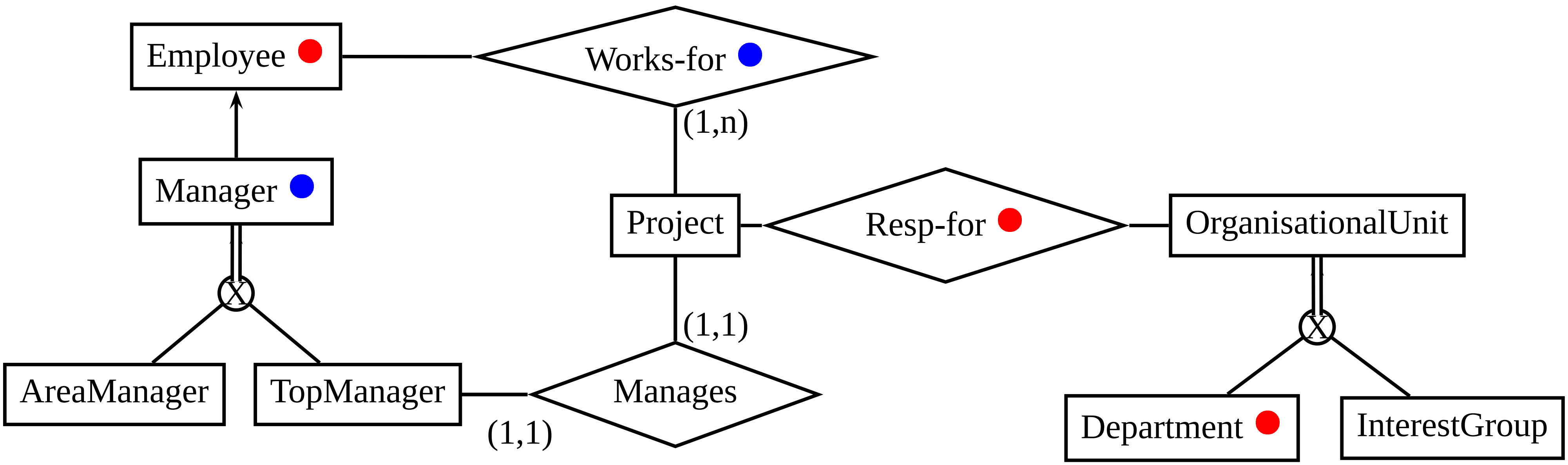
- $o \in C^{\mathcal{B}(t)} \rightarrow \forall t' \in \mathcal{T}. o \in C^{\mathcal{B}(t')}$

  $\texttt{Employee} \sqsubseteq (\Box^{+}\texttt{Employee}) \sqcap (\Box^{-}\texttt{Employee})$

- $r \in R^{\mathcal{B}(t)} \rightarrow \forall t' \in \mathcal{T}. r \in R^{\mathcal{B}(t')}$

  $\texttt{Responsible-for} \sqsubseteq (\Box^{+}\texttt{Responsible-for}) \sqcap (\Box^{-}\texttt{Responsible-for})$

A Semantics for Timestamps



- $o \in C^{\mathcal{B}(t)} \rightarrow \forall t' \in \mathcal{T}. o \in C^{\mathcal{B}(t')}$

  $\texttt{Employee} \sqsubseteq (\Box^{+}\texttt{Employee}) \sqcap (\Box^{-}\texttt{Employee})$

- $r \in R^{\mathcal{B}(t)} \rightarrow \forall t' \in \mathcal{T}. r \in R^{\mathcal{B}(t')}$

  $\texttt{Responsible-for} \sqsubseteq (\Box^{+}\texttt{Responsible-for}) \sqcap (\Box^{-}\texttt{Responsible-for})$

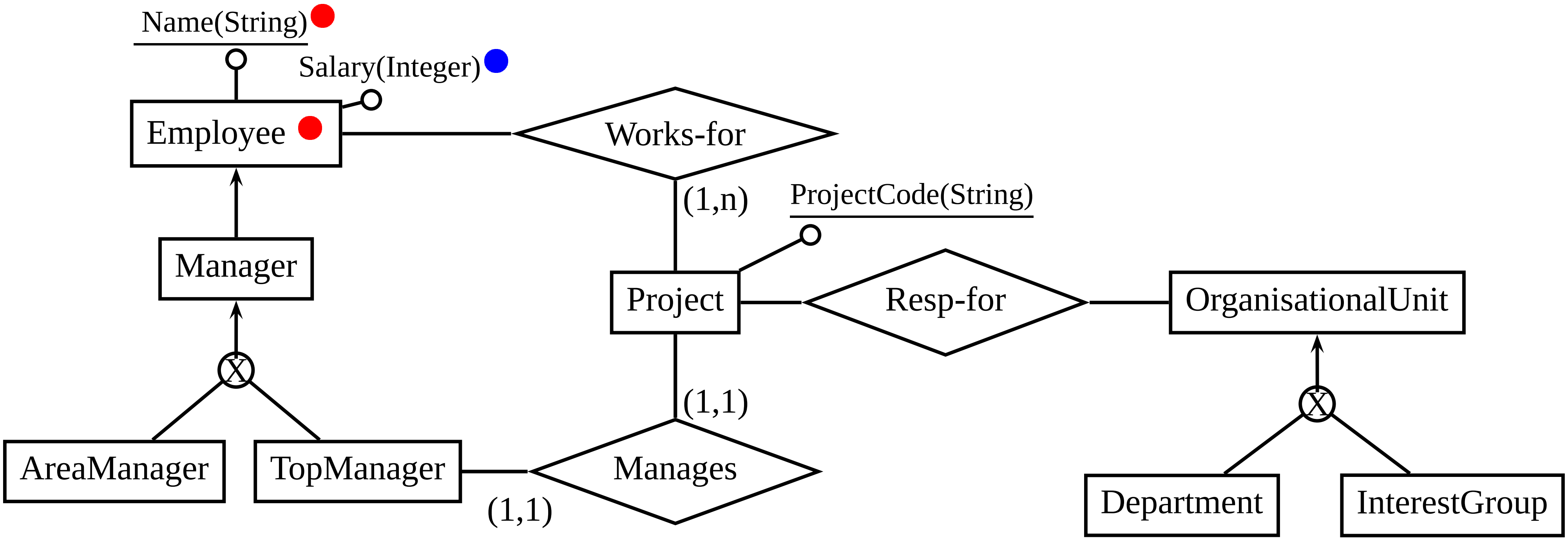
- $o \in C^{\mathcal{B}(t)} \rightarrow \exists t' \neq t. o \notin C^{\mathcal{B}(t')}$

  $\texttt{Manager} \sqsubseteq (\Diamond^{+}\neg\texttt{Manager}) \sqcup (\Diamond^{-}\neg\texttt{Manager})$

- $r \in R^{\mathcal{B}(t)} \rightarrow \exists t' \neq t. r \notin R^{\mathcal{B}(t')}$

  $\texttt{Works-for} \sqsubseteq (\Diamond^{+}\neg\texttt{Works-for}) \sqcup (\Diamond^{-}\neg\texttt{Works-for})$

Timestamping Attributes



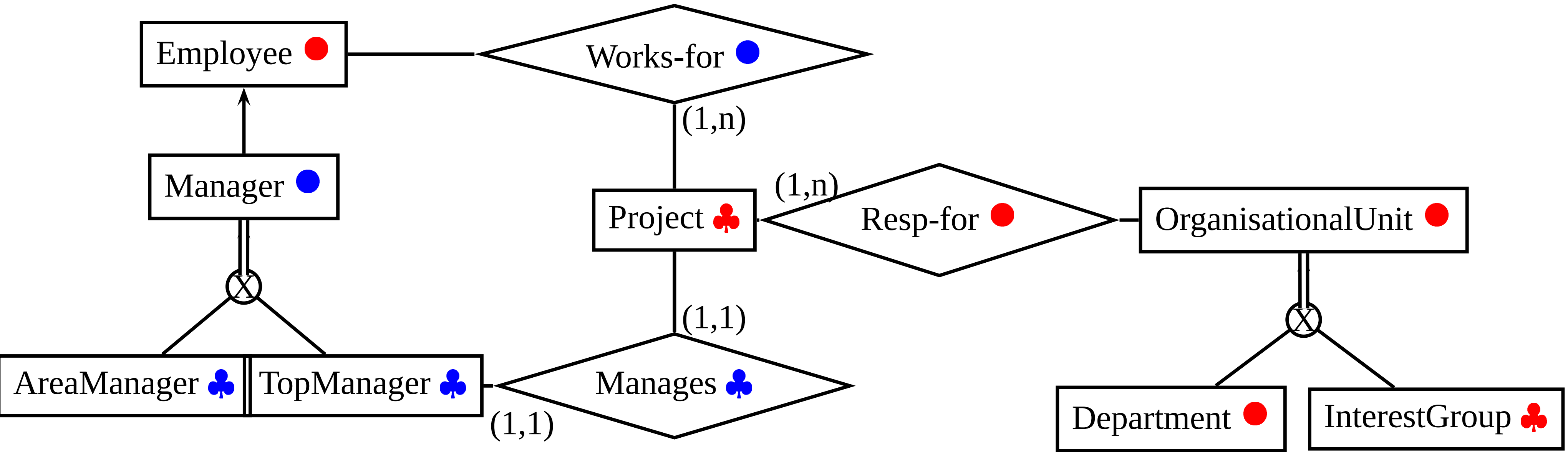
- $(o \in C^{\mathcal{B}(t)} \wedge \langle o, a_i \rangle \in A_i^{\mathcal{B}(t)}) \rightarrow \forall t' \in \mathcal{T}. \langle o, a_i \rangle \in A_i^{\mathcal{B}(t')}$

  $\mathtt{Employee} \sqsubseteq \forall \mathtt{Name.String} \sqcap (= 1 \ \mathtt{Name}) \sqcap (= 1 \ \Box^* \mathtt{Name})$

- $(o \in C^{\mathcal{B}(t)} \wedge \langle o, a_i \rangle \in A_i^{\mathcal{B}(t)}) \rightarrow \exists t' \neq t. \langle o, a_i \rangle \notin A_i^{\mathcal{B}(t')}$

  $\mathtt{Employee} \sqsubseteq \neg (= 1 \ \Box^* \mathtt{Salary})$

Logical Consequences Involving Timestamps



The following are some of the classical cases of logical implications found in the literature:

- Sub-entities of temporary entities must be temporary.
- A schema is inconsistent if exactly one of a whole set of snapshot partitioning sub-entities is temporary.
- Participants of snapshot relationships must be snapshot entities when they participate at least once.
- A relationship is temporary if one of the participating entities is temporary.

Outline

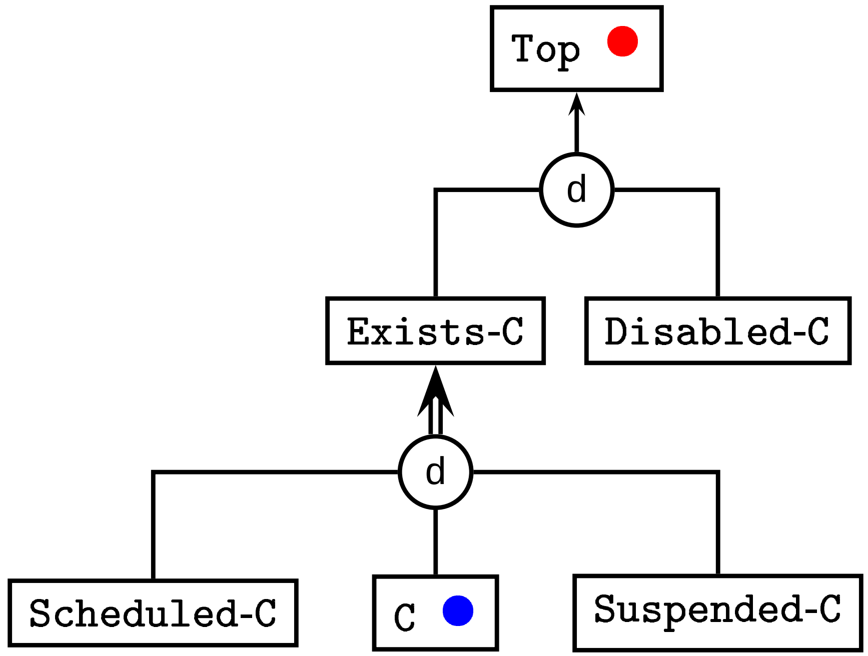
- Modeling Timestamping
- Modeling Evolution Constraints
  - Status Classes
  - Transitions
  - Generation Relationships
  - Cross-Time Relationships

Evolution Constraints: Status Classes

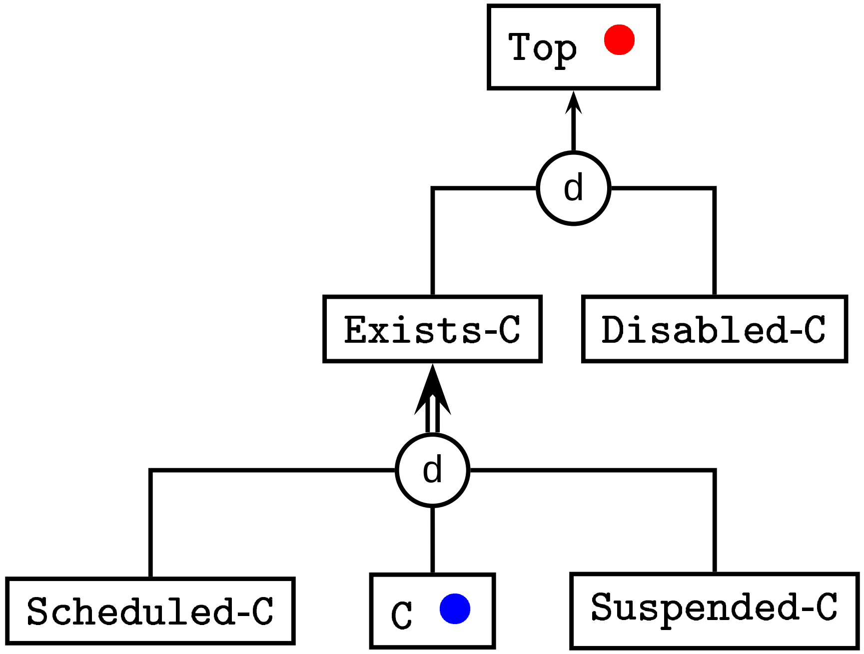
Describe the evolving status of membership of each object in the class. Four different statuses can be specified, together with precise transitions between them:

- Scheduled. An object is scheduled if its existence within the class is known but its membership in the class will only become effective some time later.
- Active. The status of an object is active if the object is a full member of the class.
- Suspended. This status qualifies objects that exist as members of the class, but are to be seen as inactive members of the class.
- Disabled. It is used to model expired objects in a class.

A Semantics for Status Classes



A Semantics for Status Classes



(EXISTS) *Existence persists until Disabled.*

$$o \in \texttt{Exists-C}^{\mathcal{B}(t)} \rightarrow \forall t' > t. (o \in \texttt{Exists-C}^{\mathcal{B}(t')} \vee o \in \texttt{Disabled-C}^{\mathcal{B}(t')})$$

$$\texttt{Exists-C} \sqsubseteq \Box^{+}(\texttt{Exists-C} \sqcup \texttt{Disabled-C})$$

(DISAB1) *Disabled persists.*

$$o \in \texttt{Disabled-C}^{\mathcal{B}(t)} \rightarrow \forall t' > t. o \in \texttt{Disabled-C}^{\mathcal{B}(t')}$$

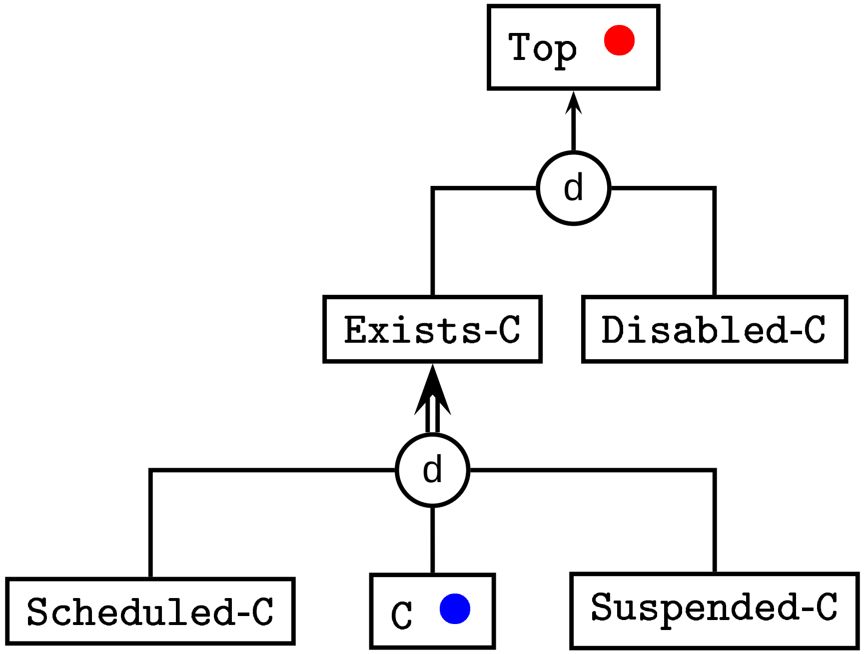
$$\texttt{Disabled-C} \sqsubseteq \Box^{+}\texttt{Disabled-C}$$

(DISAB2) *Disabled was Active in the past.*

$$o \in \texttt{Disabled-C}^{\mathcal{B}(t)} \rightarrow \exists t' < t. o \in \texttt{C}^{\mathcal{B}(t')}$$

$$\texttt{Disabled-C} \sqsubseteq \Diamond^{-}\texttt{C}$$

A Semantics for Status Classes (Cont.)



(SUSP) *Suspended was Active in the past.*

$$o \in \texttt{Suspended-C}^{\mathcal{B}(t)} \rightarrow \exists t' < t. o \in \texttt{C}^{\mathcal{B}(t')}$$

$$\texttt{Suspended-C} \sqsubseteq \Diamond^{-}\texttt{C}$$

(SCH1) *Scheduled will eventually become Active.*

$$o \in \texttt{Scheduled-C}^{\mathcal{B}(t)} \rightarrow \exists t' > t. o \in \texttt{C}^{\mathcal{B}(t')}$$

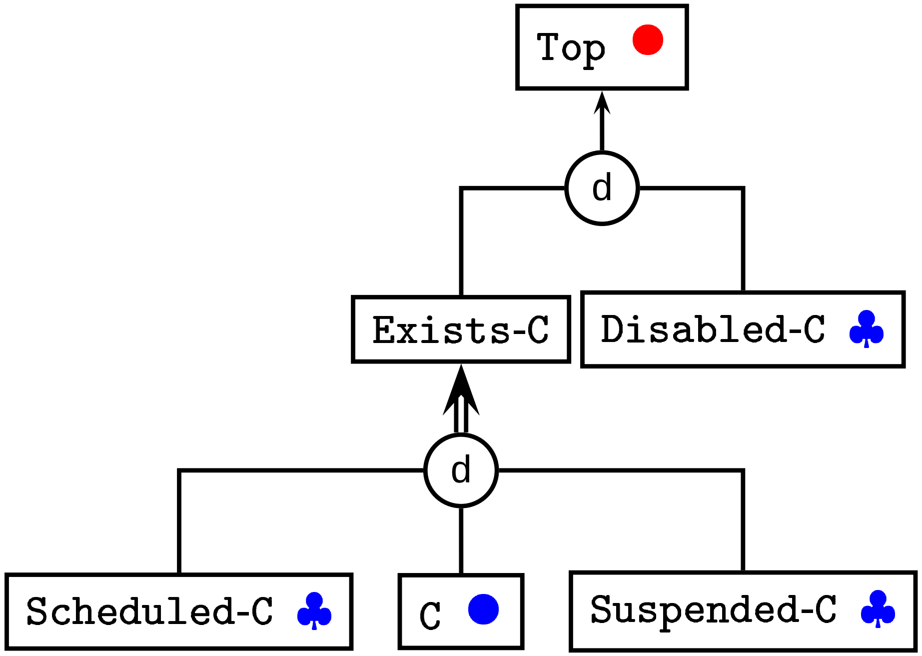
$$\texttt{Scheduled-C} \sqsubseteq \Diamond^{+}\texttt{C}$$

(SCH2) *Scheduled can never follow Active.*

$$o \in \texttt{C}^{\mathcal{B}(t)} \rightarrow \forall t' > t. o \notin \texttt{Scheduled-C}^{\mathcal{B}(t')}$$

$$\texttt{C} \sqsubseteq \Box^{+}\neg\texttt{Scheduled-C}$$

Logical Consequences from Status Classes



(TEMP) *Scheduled, Suspended and Disabled are temporary classes.*

(SCH3) *Scheduled persists until active.*

$\texttt{Scheduled-C} \sqsubseteq \texttt{Scheduled-C}\,\mathcal{U}\,\texttt{C}.$

(SCH4) *Scheduled cannot evolve directly to Disabled*

$\texttt{Scheduled-C} \sqsubseteq \oplus\neg\texttt{Disbled-C}.$

(DISAB3) *Disabled was active but it will never become active anymore*

$\texttt{Disabled-C} \sqsubseteq \Diamond^{-}(\texttt{C} \sqcap \Box^{+}\neg\texttt{C}).$

Outline

- Modeling Timestamping
- Modeling Evolution Constraints
  - Status Classes
  - Transitions
  - Generation Relationships
  - Cross-Time Relationships

Evolution Constraints: Transitions

Dynamic Transitions between classes model the notion of object migration from a source to a target class.

1. *Dynamic Evolution*, when an object ceases to be an instance of a source class;
   - **Example.** "An area manger can become a top manger while ceasing to be an area manager.".

2. *Dynamic Extension*, when an object is still allowed to belong to the source.
   - **Example.** "An employee can become a manger.".

A Semantics for Transitions

Specifying a transition between two classes means that:

1. We want to keep track of such migration;
2. Not necessarily all the objects in the source participate in the migration;
3. When the source class is a temporal class, migration involves only objects "existing" in the class (i.e., scheduled, active and suspended objects). Thus, disabled objects cannot take part in a transition.

A Semantics for Transitions (Cont.)

- We introduce two classes denoted by either $\textsc{dex}_{C_1,C_2}$ or $\textsc{dev}_{C_1,C_2}$ for dynamic extension and evolution, respectively.

- Semantics for *dynamic extension* between classes C_1, C_2.
$o \in \textsc{dex}^{\mathcal{B}(t)}_{C_1,C_2} \to (o \in \texttt{Exists-C}_1{}^{\mathcal{B}(t)} \wedge o \in \texttt{Scheduled-C}_2{}^{\mathcal{B}(t)} \wedge o \in C_2^{\mathcal{B}(t+1)})$
$\textsc{dex}_{C_1,C_2} \sqsubseteq \texttt{Exists-C}_1 \sqcap \texttt{Scheduled-C}_2 \sqcap \oplus C_2.$

- Semantics for *dynamic evolution* between classes C_1, C_2.
$o \in \textsc{dev}^{\mathcal{B}(t)}_{C_1,C_2} \to (o \in \texttt{Exists-C}_1{}^{\mathcal{B}(t)} \wedge o \in \texttt{Scheduled-C}_2{}^{\mathcal{B}(t)} \wedge o \in C_2^{\mathcal{B}(t+1)} \wedge$
$$\forall t' \geq t+1. (o \in C_2^{\mathcal{B}(t')} \to o \notin C_1^{\mathcal{B}(t')}))$$
$\textsc{dev}_{C_1,C_2} \sqsubseteq \textsc{dex}_{C_1,C_2} \sqcap \Box^{+}(C_2 \to \neg C_1)$

Logical Consequences from Transitions

1. The classes $\textsc{dex}_{C_1,C_2}$ and $\textsc{dev}_{C_1,C_2}$ are temporary classes (actually, they are instantaneous).
2. Objects in the classes $\textsc{dex}_{C_1,C_2}$ and $\textsc{dev}_{C_1,C_2}$ cannot be disabled as C_2.
3. The target class C_2 cannot be snapshot (it becomes temporary if all of its members are involved in the migration).
4. The source class C_1 cannot be snapshot when it is involved into a dynamic evolution (it becomes temporary if all of its members are involved in the migration).
5. Dynamic evolution cannot involve sub-classes (Note: this implication doesn't hold for dynamic extension).

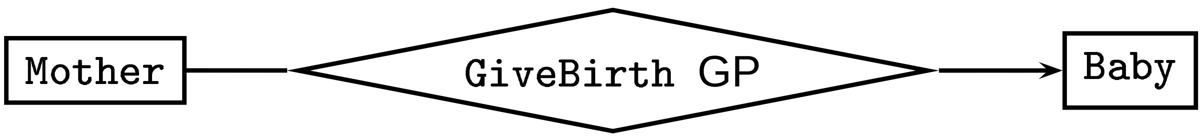
Outline

- Modeling Timestamping
- Modeling Evolution Constraints
  - Status Classes
  - Transitions
  - Generation Relationships
  - Cross-Time Relationships

Evolution Constraints: Generation Relationships

Generation relationships represent processes that lead to the emergence of new instances starting from a set of instances.

1. *Production Relationships*, when the source objects survive the generation process (GP marked).



2. *Transformation Relationships*, when all the instances involved in the process are consumed (GT marked).

A Semantics for Generation Relationships

We model generation as binary relationships connecting a source class to a target one:

$$\textsc{rel}(R) = \langle \mathtt{source} : C_1, \mathtt{target} : \mathtt{Scheduled\text{-}C_2} \rangle$$

- Semantics for Production Relationships

  $\langle o_1, o_2 \rangle \in R^{\mathcal{B}(t)} \rightarrow (o_1 \in C_1^{\mathcal{B}(t)} \wedge o_2 \in \mathtt{Scheduled\text{-}C_2}^{\mathcal{B}(t)} \wedge o_2 \in C_2^{\mathcal{B}(t+1)})$

  $R \sqsubseteq \mathtt{source} : C_1 \sqcap \mathtt{target} : (\mathtt{Scheduled\text{-}C_2} \sqcap \oplus C_2)$

- Semantics for Transformation Relationships

  $\langle o_1, o_2 \rangle \in R^{\mathcal{B}(t)} \rightarrow (o_1 \in C_1^{\mathcal{B}(t)} \wedge o_1 \in \mathtt{Disabled\text{-}C_1}^{\mathcal{B}(t+1)} \wedge$
  $o_2 \in \mathtt{Scheduled\text{-}C_2}^{\mathcal{B}(t)} \wedge o_2 \in C_2^{\mathcal{B}(t+1)})$

  $R \sqsubseteq \mathtt{source} : (C_1 \sqcap \oplus \mathtt{Disabled\text{-}C_1}) \sqcap \mathtt{target} : (\mathtt{Scheduled\text{-}C_2} \sqcap \oplus C_2)$

Logical Consequences from Generation Relationships

1. The target class, C_2, cannot be snapshot (it becomes temporary if total participation is specified).
2. A generation relationship, R, is temporary.
3. If R is a transformation relationship, then, C_1 cannot be snapshot.

Outline

- Modeling Timestamping
- Modeling Evolution Constraints
  - Status Classes
  - Transitions
  - Generation Relationships
  - Cross-Time Relationships

Evolution Constraints: Cross-Time Relationships

Cross-time relationships relate objects that are members of the participating classes at different times.

- We formalize cross-time relationships with the aim of preserving the snapshot reducibility.
- **Example:**
  - $\mathtt{Biography} \subseteq \mathtt{Author} \times \mathtt{Person}$
  - $\mathtt{bio} = \langle \mathtt{Tulard}, \mathtt{Napoleon} \rangle$ and $\mathtt{bio} \in \mathtt{Biography}^{\mathcal{B}(\mathtt{1984})}$

Evolution Constraints: Cross-Time Relationships

Cross-time relationships relate objects that are members of the participating classes at different times.

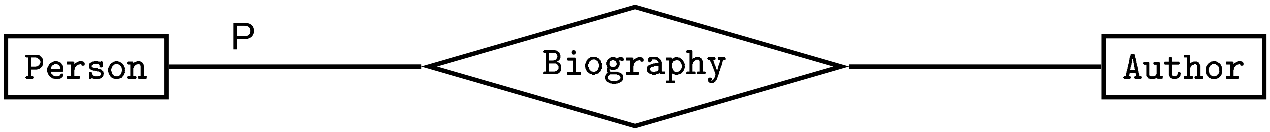
- We formalize cross-time relationships with the aim of preserving the snapshot reducibility.
- **Example:**
  - $\texttt{Biography} \subseteq \texttt{Author} \times \texttt{Person}$
  - $\texttt{bio} = \langle \texttt{Tulard}, \texttt{Napoleon} \rangle$ and $\texttt{bio} \in \texttt{Biography}^{\mathcal{B}(1984)}$
- Snapshot Reducibility would imply the following constraints:
  - $\texttt{Tulard} \in \texttt{Author}^{\mathcal{B}(1984)}$;
  - $\texttt{Napoleon} \in \texttt{Person}^{\mathcal{B}(1984)}$

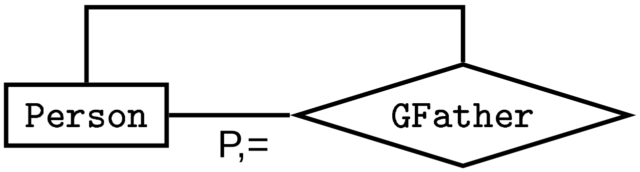
Evolution Constraints: Cross-Time Relationships

Cross-time relationships relate objects that are members of the participating classes at different times.

- We formalize cross-time relationships with the aim of preserving the snapshot reducibility.
- **Example:**
  - $\texttt{Biography} \subseteq \texttt{Author} \times \texttt{Person}$
  - $\texttt{bio} = \langle \texttt{Tulard}, \texttt{Napoleon} \rangle$ and $\texttt{bio} \in \texttt{Biography}^{\mathcal{B}(1984)}$
- Snapshot Reducibility would imply the following constraints:
  - $\texttt{Tulard} \in \texttt{Author}^{\mathcal{B}(1984)}$;
  - $\texttt{Napoleon} \in \texttt{Person}^{\mathcal{B}(1984)}$
- **Solution.** Use status classes to preserve snapshot reducibility.
  - `Napoleon` is a member of the `Disabled-Person` class in 1984.

A Semantics for Status Classes





- Strictly Past (P).

  $r = \langle e_1, e_2 \rangle \in R^{\mathcal{B}(t)} \to e_1 \in \texttt{Disabled-C}_1{}^{\mathcal{B}(t)} \wedge e_2 \in C_2^{\mathcal{B}(t)}$

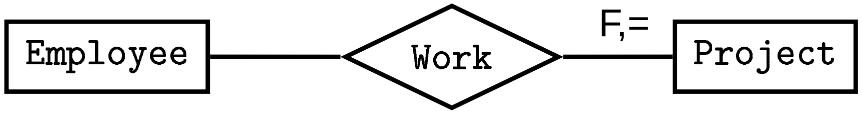
  $\texttt{R} \sqsubseteq \texttt{U}_1 : \texttt{Disabled-C}_1 \sqcap \texttt{U}_2 : \texttt{C}_2.$

- Past (P,=)

  $r = \langle e_1, e_2 \rangle \in R^{\mathcal{B}(t)} \to e_1 \in (C_1 \sqcup \texttt{Disabled-C}_1)^{\mathcal{B}(t)} \wedge e_2 \in C_2^{\mathcal{B}(t)}$

  $\texttt{R} \sqsubseteq \texttt{U}_1 : (\texttt{C}_1 \sqcup \texttt{Disabled-C}_1) \sqcap \texttt{U}_2 : \texttt{C}_2$

A Semantics for Status Classes (Cont.)



- Strictly Future (F)

  $r = \langle e_1, e_2 \rangle \in R^{\mathcal{B}(t)} \rightarrow e_1 \in \mathtt{Scheduled\text{-}C_1}^{\mathcal{B}(t)} \wedge e_2 \in C_2^{\mathcal{B}(t)}$

  $\mathtt{R} \sqsubseteq \mathtt{U_1} : \mathtt{Scheduled\text{-}C_1} \sqcap \mathtt{U_2} : \mathtt{C_2}$

- Future (F,=)

  $r = \langle e_1, e_2 \rangle \in R^{\mathcal{B}(t)} \rightarrow e_1 \in (C_1 \sqcup \mathtt{Scheduled\text{-}C_1})^{\mathcal{B}(t)} \wedge e_2 \in C_2^{\mathcal{B}(t)}$

  $\mathtt{R} \sqsubseteq \mathtt{U_1} : (\mathtt{C_1} \sqcup \mathtt{Scheduled\text{-}C_1}) \sqcap \mathtt{U_2} : \mathtt{C_2}$

Further Work

- $\mathcal{ER}_{VT}$ **Vs. Temporal DB.**
  - $\mathcal{ER}_{VT}$ with just timestamping can be translated into a relational models with Timestamps [Bassel:MSc-Thesis'02];
  - **How does the translation change in presence of evolution constraints?**
- **Reasoning.**
  - $\mathcal{ER}_{VT}$ with evolution constraints is undecidable [Artale:TIME'04];
  - $\mathcal{ER}_{VT}$ with Timestamping on Entities plus Temporal IC on Entities is decidable [Artale:et:al:JELIA'02];
  - **Does reasoning on $\mathcal{ER}_{VT}$ with full Timestamping but without Temporal IC become decidable?**
    - *Hint.* Check the decidability of the epistemic description logic S5 $\times\mathcal{ALCQI}$.