A formal comparison of conceptual data modeling languages

—A prelude to intelligent CASE tools—

C. Maria Keet

Faculty of Computer Science, Free University of Bozen-Bolzano, Italy
keet@inf.unibz.it

Knowledge Systems Group, Meraka Institute, Pretoria, July 17 2008

Presentation is based on the EMMSAD'08 presentation & paper, of which a shorter version is available as DL'08 paper:

- Keet, C.M. A formal comparison of conceptual data modeling languages. 13th International Workshop on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD'08). Montpellier, France, 16-17 June 2008. CEUR-WS Vol-337, pp25-39.
- Keet, C.M. Unifying industry-grade class-based conceptual data modeling languages with $\mathcal{CM}_{com}$. 21st International Workshop on Description Logics (DL'08), 13-16 May 2008, Dresden, Germany. CEUR-WS, Vol-353.

Long-term scopes

- Requests for automated, online, interoperability among diverse conceptual data models and compatibility between industry-grade conceptual data modeling languages.
  - (Semi-)Standards, such as Barker ER, IE, IDEF1X, and UML
  - Implementations in CASE tools, such as VisioModeler, NORMA, CaseTalk, RationalRose, VP-UML, and SmartDraw
- Interest in reasoning over conceptual models and other online usage of conceptual models is growing from the side of modelers and early-adopter industry.

What do we have?

- From conceptual modeling: diagram-based transformations between the main languages [H01]
  - **Problems**: for each new notation a new mapping scheme has to be identified, $m : n$ mesh with $(k-1)k/2$ required mappings among k languages, and informal transformations
- From computational logic: unify class-based modeling languages through the $\mathcal{DLR}$ family of Description Logic languages, avenue for *formal* 1:n mappings [CLN99]
  - **Problems**: worked out flexibly for restricted versions of ER and frame-based systems only but not full EER, UML or ORM/ORM2, and more expressive $\mathcal{DLR}$s are available now

What do we have?

- From conceptual modeling: diagram-based transformations between the main languages [H01]
    - **Problems**: for each new notation a new mapping scheme has to be identified, $m:n$ mesh with $(k-1)k/2$ required mappings among k languages, and informal transformations
- From computational logic: unify class-based modeling languages through the $\mathcal{DLR}$ family of Description Logic languages, avenue for *formal* 1:n mappings [CLN99]
    - **Problems**: worked out flexibly for restricted versions of ER and frame-based systems only but not full EER, UML or ORM/ORM2, and more expressive $\mathcal{DLR}$s are available now

Methodology

Q: What is the greatest common denominator (or core) of the industry-grade conceptual data modeling languages?

⇒ First steps: compare ER, EER, UML class diagrams, ORM, and ORM2 and identify greatest common denominator

- Extend and refine [CGLNR98, CLN98, CLN99] by
  - integrating previously obtained results on mappings between conceptual modelling languages and characteristics of the DL languages
  - taking into account standardized (UML, IDEF1X) and semi-standardized (Barker ER, IE, ORM, ORM2) languages and their implementations (a.o., VisioModeler, NORMA, CaseTalk, RationalRose, VP-UML, and SmartDraw)

Methodology

Q: What is the greatest common denominator (or core) of the industry-grade conceptual data modeling languages?

$\Rightarrow$ First steps: compare ER, EER, UML class diagrams, ORM, and ORM2 and identify greatest common denominator

- Extend and refine [CGLNR98, CLN98, CLN99] by
  - integrating previously obtained results on mappings between conceptual modelling languages and characteristics of the DL languages
  - taking into account standardized (UML, IDEF1X) and semi-standardized (Barker ER, IE, ORM, ORM2) languages and their implementations (a.o., VisioModeler, NORMA, CaseTalk, RationalRose, VP-UML, and SmartDraw)

Methodology and look ahead to results

- $\mathcal{DLR}_{ifd}$ used to formally define the generic common conceptual data modeling language $\mathcal{CM}_{com}$, i.e., with syntax and (model-theoretic) semantics
- This $\mathcal{CM}_{com}$ is used to formally define and compare ER, EER, UML class diagrams, ORM, and ORM2.
- Need to resolve main issues:
  - Establish what exactly is, or is not, part of "the" ER and EER, include textual or OCL constraints
  - Decide what to do with an officially informal conceptual modeling language (UML) or if there are alternative formalisations (ORM)

Methodology and look ahead to results

- $\mathcal{DLR}_{ifd}$ used to formally define the generic common conceptual data modeling language $\mathcal{CM}_{com}$, i.e., with syntax and (model-theoretic) semantics
- This $\mathcal{CM}_{com}$ is used to formally define and compare ER, EER, UML class diagrams, ORM, and ORM2.
- Need to resolve main issues:
  - Establish what exactly is, or is not, part of “the” ER and EER, include textual or OCL constraints
  - Decide what to do with an officially informal conceptual modeling language (UML) or if there are alternative formalisations (ORM)

Description Logics

- The basic ingredients of all DL languages are *concepts* and *roles*, where a DL-role is an *n*-ary predicate ($n \geq 2$)
- A DL language has several constructs, thereby giving greater or lesser expressivity and efficiency of automated reasoning
- DL knowledge bases are composed of the *Terminological Box* (TBox) with axioms at the concept-level, and the *Assertional Box* (ABox) with assertions about instances
- A TBox corresponds to a formal conceptual data model or can be used to represent a type-level ontology

The base language: $\mathcal{DLR}$

Take atomic relations (**P**) and atomic concepts A as the basic elements of $\mathcal{DLR}$, which allows us to construct arbitrary relations (arity ≥ 2) and arbitrary concepts according to the syntax:

$\mathbf{R} \longrightarrow \top_n | \mathbf{P} \mid (\$i/n : C) \mid \neg\mathbf{R} \mid \mathbf{R}_1 \sqcap \mathbf{R}_2$

$C \longrightarrow \top_1 | A \mid \neg C \mid C_1 \sqcap C_2 \mid \exists[\$i]\mathbf{R} \mid \leq k[\$i]\mathbf{R}$

i denotes a component of a relation; if components are not named, then integer numbers between 1 and n_{max} are used, where n is the arity of the relation. Only relations of the same arity can be combined to form expressions of type $\mathbf{R}_1 \sqcap \mathbf{R}_2$, and $i \leq n$

The base language: $\mathcal{DLR}$

The model-theoretic semantics of $\mathcal{DLR}$ is specified through the usual notion of interpretation, where $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, and the interpretation function $\cdot^{\mathcal{I}}$ assigns to each concept C a subset $C^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ and to each n-ary $\mathbf{R}$ a subset $\mathbf{R}^{\mathcal{I}}$ of $(\Delta^{\mathcal{I}})^n$, such that the conditions are satisfied following:

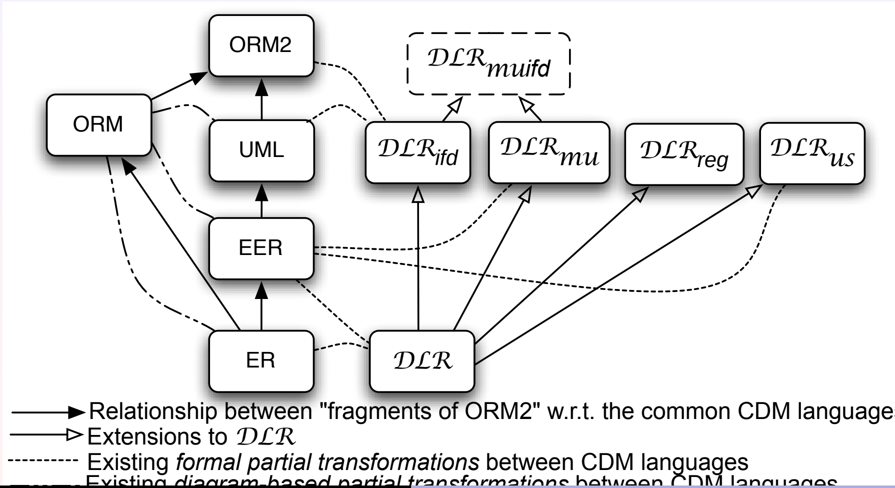
$$\top_n^{\mathcal{I}} \subseteq (\Delta^{\mathcal{I}})^n$$
$$\mathbf{P}^{\mathcal{I}} \subseteq \top_n^{\mathcal{I}}$$
$$(\neg \mathbf{R})^{\mathcal{I}} = \top_n^{\mathcal{I}} \setminus \mathbf{R}^{\mathcal{I}}$$
$$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$$
$$\top_1^{\mathcal{I}} = \Delta^{\mathcal{I}}$$

$$(\mathbf{R}_1 \sqcap \mathbf{R}_2)^{\mathcal{I}} = \mathbf{R}_1^{\mathcal{I}} \cap \mathbf{R}_2^{\mathcal{I}}$$
$$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$$
$$(C_1 \sqcap C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$$
$$(\$i/n : C)^{\mathcal{I}} = \{(d_1, ..., d_n) \in \top_n^{\mathcal{I}} | d_i \in C^{\mathcal{I}}\}$$
$$(\exists[\$i]\mathbf{R})^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} | \exists (d_1, ..., d_n) \in \mathbf{R}^{\mathcal{I}}.d_i = d\}$$
$$(\leq k[\$i]\mathbf{R})^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} || \{(d_1, ..., d_n) \in \mathbf{R}_1^{\mathcal{I}} | d_i = d | \} \leq k$$

The base language: $\mathcal{DLR}$

A knowledge base is a finite set $\mathcal{KB}$ of $\mathcal{DLR}$ (or $\mathcal{DLR}_{ifd}$) axioms of the form $C_1 \sqsubseteq C_2$ and $R_1 \sqsubseteq R_2$.
An interpretation $\mathcal{I}$ satisfies $C_1 \sqsubseteq C_2$ ($R_1 \sqsubseteq R_2$) if and only if the interpretation of C_1 (R_1) is included in the interpretation of C_2 (R_2), i.e. $C_1^{\mathcal{I}(t)} \subseteq C_2^{\mathcal{I}(t)}$ ($R_1^{\mathcal{I}(t)} \subseteq R_2^{\mathcal{I}(t)}$).
$\top_1$ denotes the interpretation domain, $\top_n$ for $n \geq 1$ denotes a subset of the n-cartesian product of the domain, which covers all introduced n-ary relations.
($\$i/n : C$) denotes all tuples in $\top_n$ that have an instance of C as their i-th component.

Relations between the 5 $\mathcal{DLR}$s



$\mathcal{DLR}_{ifd}$

- $\mathcal{DLR}_{ifd}$ has two additional constructs compared to $\mathcal{DLR}$:
  - *i*dentification assertions on a concept C, which has the form $(\textbf{id}\ C[i_1]R_1, ..., [i_h]R_h)$, where each R_j is a relation and each i_j denotes one component of R_j.
  - Non-unary *f*unctional *d*ependency assertions on a relation R, which has the form $(\textbf{fd}\ R\ i_1, ..., i_h \rightarrow j)$, where $h \geq 2$, and $i_1, ..., i_h, j$ denote components of R
- Syntax and semantics as for $\mathcal{DLR}$

$\mathcal{CM}_{com}$ syntax

Definition (Conceptual Data Model $\mathcal{CM}_{com}$ syntax)

A $\mathcal{CM}_{com}$ conceptual data model is a tuple $\Sigma = (\mathcal{L}, \text{REL}, \text{ATT}, \text{CARD}, \text{ISA}_C, \text{ISA}_R, \text{ISA}_U, \text{DISJ}, \text{COVER}, \text{KEY}, \text{EXTK}, \text{FD}, \text{OBJ}, \text{REX}, \text{RDM})$ such that:

- $\mathcal{L}$ is a finite alphabet partitioned into the sets: $\mathcal{C}$ (*class* symbols), $\mathcal{A}$ (*attribute* symbols), $\mathcal{R}$ (*relationship* symbols), $\mathcal{U}$ (*role* symbols), and $\mathcal{D}$ (*domain* symbols); the tuple $(\mathcal{C}, \mathcal{A}, \mathcal{R}, \mathcal{U}, \mathcal{D})$ is the *signature* of the conceptual data model Σ.
- REL is a function that maps a relationship symbol in $\mathcal{R}$ to an $\mathcal{U}$-labeled tuple over $\mathcal{C}$, $\text{REL}(R) = \langle U_1 : C_1, \ldots, U_k : C_k \rangle$, and k is the *arity* of R.
- ...

Example: syntax for $\mathcal{CM}_{com}$

- ISA for, e.g., Author ISA Person
- cardinality constrains, $\mathtt{CARD}(\mathtt{Author}, \mathtt{Writes}, \mathtt{auth}) = (1, n)$
- DISJ and COVER where $\{\mathtt{Author}, \mathtt{Editor}\}$ DISJ Person and $\{\mathtt{Author}, \mathtt{Editor}\}$ COVER Person
- $\mathtt{KEY}(\mathtt{Person}) = \mathtt{id}$
- Equivalent representation in $\mathcal{DLR}_{ifd}$ as: $\mathtt{Author} \sqsubseteq \mathtt{Person}$ (subsumption), $\mathtt{Author} \sqsubseteq \exists[\mathtt{auth}]\mathtt{writes}$ (at least one), $\mathtt{Author} \sqsubseteq \neg\mathtt{Editor}$ (disjoint), $\mathtt{Person} \sqsubseteq \mathtt{Author} \sqcup \mathtt{Editor}$ (covering), and $\mathtt{Person} \sqsubseteq \exists^{=1}[\mathtt{From}]\mathtt{id}$, $\top \sqsubseteq \exists^{\leq 1}[\mathtt{To}](\mathtt{id} \sqcap [\mathtt{From}] : \mathtt{Person})$ (key)

Example: syntax for $\mathcal{CM}_{com}$

- ISA for, e.g., Author ISA Person
- cardinality constrains, $\text{CARD}(\texttt{Author}, \texttt{Writes}, \texttt{auth}) = (1, n)$
- DISJ and COVER where $\{\texttt{Author}, \texttt{Editor}\}$ DISJ Person and $\{\texttt{Author}, \texttt{Editor}\}$ COVER Person
- $\text{KEY}(\texttt{Person}) = \texttt{id}$

- Equivalent representation in $\mathcal{DLR}_{ifd}$ as: $\texttt{Author} \sqsubseteq \texttt{Person}$ (subsumption), $\texttt{Author} \sqsubseteq \exists[\texttt{auth}]\texttt{writes}$ (at least one), $\texttt{Author} \sqsubseteq \neg\texttt{Editor}$ (disjoint), $\texttt{Person} \sqsubseteq \texttt{Author} \sqcup \texttt{Editor}$ (covering), and $\texttt{Person} \sqsubseteq \exists^{=1}[\texttt{From}]\texttt{id}$, $\top \sqsubseteq \exists^{\leq 1}[\texttt{To}](\texttt{id} \sqcap [\texttt{From}] : \texttt{Person})$ (key)

Example: syntax for $\mathcal{CM}_{com}$

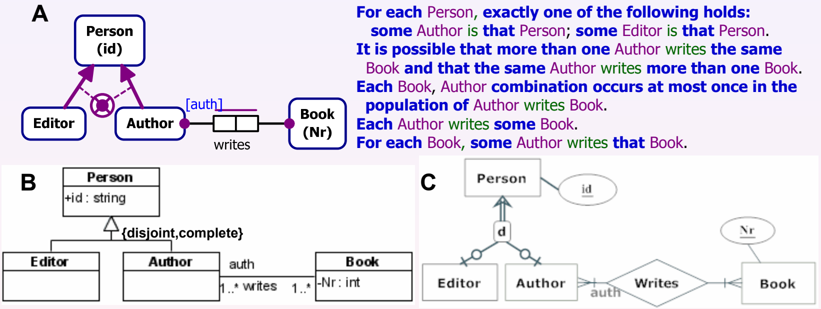


Figure: Examples of graphical syntax for $\mathcal{CM}_{com}$ with ORM2 drawn in NORMA (A), UML class diagram drawn in VP-UML (B), and EER drawn with SmartDraw (C).

$\mathcal{CM}_{com}$ semantics

Definition ($\mathcal{CM}_{com}$ Semantics)

Let Σ be a $\mathcal{CM}_{com}$ conceptual data model. An *interpretation* for the conceptual model Σ is a tuple $\mathcal{B} = (\Delta^{\mathcal{B}} \cup \Delta^{\mathcal{B}}_{D}, \cdot^{\mathcal{B}})$, such that:

- $\Delta^{\mathcal{B}}$ is a nonempty set of abstract objects disjoint from $\Delta^{\mathcal{B}}_{D}$;
- $\Delta^{\mathcal{B}}_{D} = \bigcup_{D_i \in \mathcal{D}} \Delta^{\mathcal{B}}_{D_i}$ is the set of basic domain values used in Σ; and
- $\cdot^{\mathcal{B}}$ is a function that maps:
    - Every basic domain symbol $D \in \mathcal{D}$ into a set $D^{\mathcal{B}} = \Delta^{\mathcal{B}}_{D_i}$.
    - ...
    - Every attribute $A \in \mathcal{A}$ to a set $A^{\mathcal{B}} \subseteq \Delta^{\mathcal{B}} \times \Delta^{\mathcal{B}}_{D}$, such that, for each $C \in \mathcal{C}$, if $\textsc{att}(C) = \langle A_1 : D_1, \ldots, A_h : D_h \rangle$, then, $o \in C^{\mathcal{B}} \rightarrow (\forall i \in \{1, \ldots, h\}, \exists a_i. \langle o, a_i \rangle \in A^{\mathcal{B}}_i \wedge \forall a_i. \langle o, a_i \rangle \in A^{\mathcal{B}}_i \rightarrow a_i \in \Delta^{\mathcal{B}}_{D_i})$.

Definition ($\mathcal{CM}_{com}$ Semantics *cont'd*)

$\mathcal{B}$ is said a *legal database state* or *legal application software state* if it satisfies all of the constraints expressed in the conceptual data model:

- For each $C_1, C_2 \in \mathcal{C}$: if C_1 ISA_C C_2, then $C_1^{\mathcal{B}} \subseteq C_2^{\mathcal{B}}$.
- For each $R_1, R_2 \in \mathcal{R}$: if R_1 ISA_R R_2, then $R_1^{\mathcal{B}} \subseteq R_2^{\mathcal{B}}$.
- For each $U_1, U_2 \in \mathcal{U}$, $R_1, R_2 \in \mathcal{R}$, $\text{REL}(R_1) = \langle U_1 : o_1, \ldots, U_n : o_n \rangle$, $\text{REL}(R_2) = \langle U_2 : o_2, \ldots, U_m : o_m \rangle$, $n = m$, $R_1 \neq R_2$: if U_1 ISA_U U_2, then $U_1^{\mathcal{B}} \subseteq U_2^{\mathcal{B}}$.
- ...

Definition ($\mathcal{CM}_{com}$ Semantics *cont'd*)

- For each $C \in \mathcal{C}$, $R_h \in \mathcal{R}$, $h \geq 1$, $\text{REL}(R_h) = \langle U : C, U_1 : C_1, \ldots, U_k : C_k \rangle$, $k \geq 1$, $k + 1$ the arity of R_h, such that $\text{EXTK}(C) = [U_1]R_1, \ldots, [U_h]R_h$, then for all $o_a, o_b \in C^{\mathcal{B}}$ and for all $t_1, s_1 \in R_1^{\mathcal{B}}$, ..., $t_h, s_h \in R_h^{\mathcal{B}}$ we have that:

$$\left.\begin{array}{r} o_a = t_1[U_1] = ... = t_h[U_h] \\ o_b = s_1[U_1] = ... = s_h[U_h] \\ t_j[U] = s_j[U], \text{ for } j \in \{1, ..., h\}, \text{ and for } U \neq j \end{array}\right\} \quad \text{implies } o_a = o_b$$

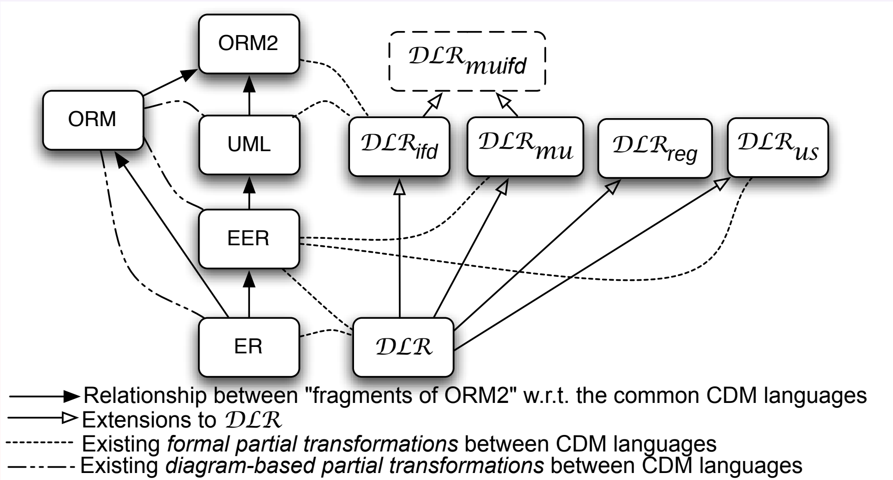
where o_a is an instance of C that is the U_j-th component of a tuple t_j of R_j, for $j \in \{1, ..., h\}$, and o_b is an instance of C that is the U_j-th component of a tuple s_j of R_j, for $j \in \{1, ..., h\}$, and for each j, t_j agrees with s_j in all components different from U_j, ...

Definition ($\mathcal{CM}_{com}$ Semantics *cont'd*)

..., then o_a and o_b are the same object.

- For each $R \in \mathcal{R}$, $U_i, j \in \mathcal{U}$, for $i \geq 2$, $i \neq j$, $\texttt{REL}(R) = \langle U_1 : C_1, \ldots, U_i : C_i, j : C_j \rangle$, $\texttt{FD}(R) = \langle U_1, \ldots, U_i \rightarrow j \rangle$, then for all $t, s \in R^{\mathcal{B}}$, we have that $t[U_1] = s[U_1], ..., t[U_i] = s[U_i]$ implies $t_j = s_j$.
- ...
- For each $U_i \in \mathcal{U}$, $i \geq 2$, $R_i \in \mathcal{R}$, each R_i has the same arity m (with $m \geq 2$), $C_j \in \mathcal{C}$ with $2 \leq j \leq i(m-1)+1$, and $\texttt{REL}(R_i) = \langle U_i : C_i, \ldots U_m : C_m \rangle$ (and, thus, $R_i \in R_i^{\mathcal{B}}$ and $o_j \in C_j^{\mathcal{B}}$), if $\{U_1, U_2, \ldots U_{i-1}\}$ $\texttt{REX}$ U_i, then $\forall i \in \{1, \ldots, i\}.o_j \in C_j^{\mathcal{B}} \rightarrow \texttt{CMIN}(o_j, r_i, u_i) \leq 1 \wedge u_i \neq u_1 \wedge \ldots \wedge u_i \neq u_{i-1}$ where $u_i \in U_i^{\mathcal{B}}$, $r_i \in R_i^{\mathcal{B}}$.

Overview


ORM2
$\mathcal{DLR}_{muifd}$
ORM
UML
$\mathcal{DLR}_{ifd}$
$\mathcal{DLR}_{mu}$
$\mathcal{DLR}_{reg}$
$\mathcal{DLR}_{us}$
EER
ER
$\mathcal{DLR}$
Relationship between "fragments of ORM2" w.r.t. the common CDM languages
Extensions to $\mathcal{DLR}$
Existing *formal partial transformations* between CDM languages
Existing *diagram-based partial transformations* between CDM languages

Definition ($\mathcal{CM}_{ER}$)

A $\mathcal{CM}_{ER}$ conceptual data model is a tuple
$\Sigma = (\mathcal{L}, \text{REL}, \text{ATT}, \text{CARD}^{-}, \text{KEY}, \text{EXTK})$
adhering to $\mathcal{CM}_{com}$ syntax and semantics except that CARD is restricted to any of the values $\{\geq 0, \leq 1, \geq 1\}$, denoted in Σ with CARD^{-}.

Definition ($\mathcal{CM}_{EER}$)

A $\mathcal{CM}_{EER}$ conceptual data model is a tuple
$\Sigma = (\mathcal{L}, \text{REL}, \text{ATT}, \text{CARD}, \text{ISA}_{C}, \text{DISJ}, \text{COVER}, \text{KEY}, \text{EXTK})$
adhering to $\mathcal{CM}_{com}$ syntax and semantics.

Definition ($\mathcal{CM}_{UML}$)

A $\mathcal{CM}_{UML}$ conceptual data model is a tuple
$\Sigma = (\mathcal{L}, \text{REL}, \text{ATT}, \text{CARD}, \text{ISA}_C, \text{ISA}_R, \text{DISJ}, \text{COVER}, \text{KEY}, \text{EXTK}, \text{FD}, \text{OBJ}, \text{PW})$
adhering to $\mathcal{CM}_{com}$ syntax and semantics, except for the aggregation association PW, with syntax
$\text{PW} = \langle U_1 : C_1, U_2 : C_2 \rangle$, that has no defined semantics.

Definition ($\mathcal{CM}_{ORM}$)

A $\mathcal{CM}_{ORM}$ conceptual data model is a tuple
$\Sigma = (\mathcal{L}, \text{REL}, \text{ATT}, \text{CARD}, \text{ISA}_C, \text{ISA}_R, \text{ISA}_U, \text{KEY}, \text{EXTK}, \text{FD}, \text{OBJ}, \text{REX}, \text{RDM}, \text{JOIN}, \text{KROL}, \text{RING}^-)$
adhering to $\mathcal{CM}_{com}$ syntax and semantics, and, in addition, such that:

- JOIN comprises the following constraints: {*join-subset, join-equality, join-exclusion*} over ≥ 2 n-ary relations, $n \geq 2$, as defined in [H89].
- KROL comprises the following constraints: {*subset over k roles, multi-role frequency, set-equality over k roles, role exclusion over k roles*} over an n-ary relation, $n \geq 3$, and $k < n$, as defined in [H89].
- RING$^-$ comprises the following constraints: {*intransitive, irreflexive, asymmetric*}, as defined in [H89].

Definition ($\mathcal{CM}_{ORM2}$)

A $\mathcal{CM}_{ORM2}$ conceptual data model is a tuple
$\Sigma = (\mathcal{L}, \text{REL}, \text{ATT}, \text{CARD}, \text{ISA}_C, \text{ISA}_R, \text{ISA}_U, \text{DISJ}, \text{COVER}, \text{KEY}, \text{EXTK}, \text{FD}, \text{OBJ}, \text{REX}, \text{RDM}, \text{JOIN}, \text{KROL}, \text{RING})$
adhering to the syntax and semantics as defined for $\mathcal{CM}_{com}$, and such that:

- KROL and JOIN are as in Definition 9.
- RING comprises the following constraints: {*intransitive, irreflexive, asymmetric, antisymmetric, acyclic, symmetric*}, as defined in [H89, H01].

Discussion

- Comparison trivial (almost) with the 5 definitions
- Four finer-grained issues
  - With ORM formalization of [H89], $\mathcal{CM}_{UML}$ not a proper fragment of $\mathcal{CM}_{ORM}$ (total exclusive subtypes–but OCL). $\mathcal{CM}_{UML}$ fragment of $\mathcal{CM}_{ORM2}$ (dismiss PW)
  - KEY is for single attribute keys (± ORM reference scheme), EXTK for multiple-attribute keys, no enforcing of elementary fact type
  - Attributes (UML, ER, EER) vs. attribute-free (ORM, ORM2). Semantics of ATT, an n-ary relation with as range(s) data type(s)
  - Some features of ORM and ORM2 missing in $\mathcal{CM}_{com}$ ...

Discussion

- Comparison trivial (almost) with the 5 definitions
- Four finer-grained issues
  - With ORM formalization of [H89], $\mathcal{CM}_{UML}$ not a proper fragment of $\mathcal{CM}_{ORM}$ (total exclusive subtypes–but OCL). $\mathcal{CM}_{UML}$ fragment of $\mathcal{CM}_{ORM2}$ (dismiss PW)
  - KEY is for single attribute keys ($\pm$ ORM reference scheme), EXTK for multiple-attribute keys. no enforcing of elementary fact type
  - Attributes (UML, ER, EER) vs. attribute-free (ORM, ORM2). Semantics of ATT, an n-ary relation with as range(s) data type(s)
  - Some features of ORM and ORM2 missing in $\mathcal{CM}_{com}$...

Discussion

- Why a comparison with $\mathcal{DLR}_{ifd}$ and $\mathcal{CM}_{com}$ and not FOL?
- DLs are well-studied FOL fragments, and by looking at (non-) correspondences, one gains better insight in properties of CM languages as well
  - $\mathcal{CM}_{UML}$, $\mathcal{CM}_{ER}$, and $\mathcal{CM}_{EER}$ are in ExpTime-complete ($\mathcal{DLR}_{ifd}$ is)
  - Knowledge about computationally more appealing fragments in NP or NLogSpace [ACKRZ07, KS06, SCS07]

Discussion

- Why a comparison with $\mathcal{DLR}_{ifd}$ and $\mathcal{CM}_{com}$ and not FOL?
- DLs are well-studied FOL fragments, and by looking at (non-) correspondences, one gains better insight in properties of CM languages as well
  - $\mathcal{CM}_{UML}$, $\mathcal{CM}_{ER}$, and $\mathcal{CM}_{EER}$ are in ExpTime-complete ($\mathcal{DLR}_{ifd}$ is)
  - Knowledge about computationally more appealing fragments in NP or NLogSpace [ACKRZ07, KS06, SCS07]

Features [KR07]

Language ⇒ **Feature** ⇓	OWL Lite	OWL DL	OWL v1.1	*DL-Lite* $\mathcal{F}$	*DL-Lite* $\mathcal{R}$	*DL-Lite* $\mathcal{A}$	$\mathcal{DLR}$ *ifd*	$\mathcal{DLR}$ μ	$\mathcal{DLR}$ *reg*
Role hierarchy	+	+	+	-	+	+	+	+	+
N-ary roles (where $n \geq 2$)	-	-	-	$\pm$	$\pm$	$\pm$	+	+	+
Role concatenation	-	-	+	-	-	-	-	-	+
Role acyclicity	-	-	-	-	-	-	-	+	-
Symmetry	+	+	+	-	+	+	-	-	-
Role values	-	-	-	-	-	+	-	-	-
Qualified number restrictions	-	-	+	-	-	-	+	+	+
One-of, enumerated classes	-	+	+	-	-	-	-	-	-
Functional dependency	+	+	+	+	-	+	+	-	+
Covering constraint over concepts	-	+	+	-	-	-	+	+	+
Complement of concepts	-	+	+	+	+	+	+	+	+
Complement of roles	-	-	+	+	+	+	+	+	+
Concept identification	-	-	-	-	-	-	+	-	-
Range typing	-	+	+	-	+	+	+	+	+
Reflexivity *	-	-	+	-	-	-	-	+	+
Antisymmetry *	-	-	-	-	-	-	-	-	-
Transitivity * ‡	+	+	+	-	-	-	-	+	+
Asymmetry ‡	+	+	+	-	+	+	-	$\pm$	-
Irreflexivity ‡	-	-	+	-	-	-	-	+	-

Features [KR07]

Language ⇒ Feature ⇓	OWL Lite	OWL DL	OWL v1.1	DL-Lite $\mathcal{F}$	DL-Lite $\mathcal{R}$	DL-Lite $\mathcal{A}$	$\mathcal{DLR}$ *ifd*	$\mathcal{DLR}$ μ	$\mathcal{DLR}$ *reg*
Role hierarchy	+	+	+	-	+	+	+	+	+
N-ary roles (where $n \geq 2$)	-	-	-	±	±	±	+	+	+
Role concatenation	-	-	+	-	-	-	-	-	+
Role acyclicity	-	-	-	-	-	-	-	+	-
Symmetry	+	+	+	-	+	+	-	-	-
Role values	-	-	-	-	-	+	-	-	-
Qualified number restrictions	-	-	+	-	-	-	+	+	+
One-of, enumerated classes	-	+	+	-	-	-	-	-	-
Functional dependency	+	+	+	+	-	+	+	-	+
Covering constraint over concepts	-	+	+	-	-	-	+	+	+
Complement of concepts	-	+	+	+	+	+	+	+	+
Complement of roles	-	-	+	+	+	+	+	+	+
Concept identification	-	-	-	-	-	-	+	-	-
Range typing	-	+	+	-	+	+	+	+	+
Reflexivity *	-	-	+	-	-	-	-	+	+
Antisymmetry *	-	-	-	-	-	-	-	-	-
Transitivity * ‡	+	+	+	-	-	-	-	+	+
Asymmetry ‡	+	+	+	-	+	+	-	±	-
Irreflexivity ‡	-	-	+	-	-	-	-	+	-

Restrictions on $\mathcal{DLR}_{\mu ifd}$ (tentative)

THEOREM

Given a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{F})$ *of* $\mathcal{DLR}_{\mu ifd}$*, where* $\mathcal{DLR}_{\mu ifd} = (\mathcal{DLR}_{ifd}, \mathcal{DLR}_{\mu})$*, satisfiability and logical implication* $\mathcal{DLR}_{\mu ifd}$ *is ExpTime-complete, provided the following conditions are met:*

- *Least (greatest) fixpoint* $\mu X.C$ ($\nu X.C$) *is used only with binary roles* $R_b \in \mathcal{R}$*;*
- R_b *does not occur in any identification assertion, i.e., for* (**id** $C[i_1]R_1, ..., [i_h]R_h$) *then* $R_b \neq R_1, ..., R_b \neq R_h$.

Discussion

- Why a comparison with $\mathcal{DLR}_{ifd}$ and $\mathcal{CM}_{com}$ and not FOL?
- DLs are well-studied FOL fragments, and by looking at (non-) correspondences, one gains better insight in properties of CM languages as well
  - $\mathcal{CM}_{UML}$, $\mathcal{CM}_{ER}$, and $\mathcal{CM}_{EER}$ are in ExpTime-complete ($\mathcal{DLR}_{ifd}$ is)
  - Knowledge about computationally more appealing fragments in NP or NLogSpace [ACKRZ07, KS06, SCS07]
- $\mathcal{DLR}_{ifd}$ most expressive common denominator; thus far, best trade-off expressiveness & computation
- With unambiguously defined syntax and semantics, modelers can keep using their preferred diagram-based language (or make a new one), have common language at the “interchange” automated transformations

Discussion

- Why a comparison with $\mathcal{DLR}_{ifd}$ and $\mathcal{CM}_{com}$ and not FOL?
- DLs are well-studied FOL fragments, and by looking at (non-) correspondences, one gains better insight in properties of CM languages as well
  - $\mathcal{CM}_{UML}$, $\mathcal{CM}_{ER}$, and $\mathcal{CM}_{EER}$ are in ExpTime-complete ($\mathcal{DLR}_{ifd}$ is)
  - Knowledge about computationally more appealing fragments in NP or NLogSpace [ACKRZ07, KS06, SCS07]
- $\mathcal{DLR}_{ifd}$ most expressive common denominator; thus far, best trade-off expressiveness & computation
- With unambiguously defined syntax and semantics, modelers can keep using their preferred diagram-based language (or make a new one), have common language at the “interchange” automated transformations

Discussion

- Why a comparison with $\mathcal{DLR}_{ifd}$ and $\mathcal{CM}_{com}$ and not FOL?
- DLs are well-studied FOL fragments, and by looking at (non-) correspondences, one gains better insight in properties of CM languages as well
  - $\mathcal{CM}_{UML}$, $\mathcal{CM}_{ER}$, and $\mathcal{CM}_{EER}$ are in ExpTime-complete ($\mathcal{DLR}_{ifd}$ is)
  - Knowledge about computationally more appealing fragments in NP or NLogSpace [ACKRZ07, KS06, SCS07]
- $\mathcal{DLR}_{ifd}$ most expressive common denominator; thus far, best trade-off expressiveness & computation
- With unambiguously defined syntax and semantics, modelers can keep using their preferred diagram-based language (or make a new one), have common language at the “interchange” automated transformations

Conclusions and current work

- ER, EER, UML class diagrams, and ORM are different proper fragments of ORM2
- ER, EER, UML class diagrams are in ExpTime
- Results obtained with $\mathcal{CM}_{com}$ simplifies (semi-)automated interoperability of conceptual data models in different graphical languages

- "$\mathcal{DLR}_{\mu ifd}$" for ORM's ring constraints, temporal extensions with $\mathcal{DLR}_{\mathcal{US}}$ and $\mathcal{ER}_{VT}$

Conclusions and current work

- ER, EER, UML class diagrams, and ORM are different proper fragments of ORM2
- ER, EER, UML class diagrams are in ExpTime
- Results obtained with $\mathcal{CM}_{com}$ simplifies (semi-)automated interoperability of conceptual data models in different graphical languages

- "$\mathcal{DLR}_{\mu ifd}$" for ORM's ring constraints, temporal extensions with $\mathcal{DLR}_{\mathcal{US}}$ and $\mathcal{ER}_{VT}$

- Artale, A., Calvanese, D., Kontchakov, R., Ryzhikov, V., Zakharyaschev, M. Reasoning over Extended ER Models. *Proc. of ER'07*, Springer LNCS 4801, 277-292.
- Calvanese, D., De Giacomo, G., Lenzerini, M., Nardi, D., Rosati, R. Description logic framework for information integration. In *Proc. of KR'98*, 2-13.
- Calvanese, C., De Giacomo, G., Lenzerini, M. On the decidability of query containment under constraints. In: *Proc. of PODS'98*, 149-158, 1998.
- Calvanese, D., Lenzerini, M., Nardi, D. Description logics for conceptual data modeling. In: Chomicki, J., Saake, G. (Eds.), *Logics for Databases and Information Systems*. Kluwer, Amsterdam. 1998.
- Calvanese, D., Lenzerini, M. & Nardi, D. Unifying class-based representation formalisms. *JAIR*, 11:199-240, 1999.
- Calvanese, D. & De Giacomo, G. Expressive description logics. In: *The DL Handbook: Theory, Implementation and Applications*, Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (Eds). Cambridge University Press, 2003. pages 178-218.
- Halpin, T.A. *A logical analysis of information systems: static aspects of the data-oriented perspective*. PhD Thesis, University of Queensland, Australia. 1989.

Halpin, T. *Information Modeling and Relational Databases*. San Francisco: Morgan Kaufmann Publishers, 2001.

Kaneiwa, K., Satoh, K. Consistency Checking Algorithms for Restricted UML Class Diagrams. In: *Proc. of FoIKS '06*, Springer LNCS 3861, 2006. 219-239.

Keet, C.M. and Rodríguez, M. Toward using biomedical ontologies: trade-offs between ontology languages. *AAAI 2007 Workshop on Semantic e-Science (SeS'07)*, 23 July 2007, Vancouver, Canada. AAAI 2007 TR WS-07-11, 65-68.

Smaragdakis, Y., Csallner, C., Subramanian, R. Scalable automatic test data generations from modeling diagrams. In: *Proc. of ASE'07*, Nov. 5-9, Atlanta, Georgia, USA. 4-13.

Thank you for your attention