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. Unifying industry-grade class-based conceptual data modeling languages with $CM_{com}$. 21st International Workshop on Description Logics (DL’08), 13-16 May 2008, Dresden, Germany. CEUR-WS, Vol-353.
1 Background
   - Motivation
   - Methodology

2 The $DLR$ family
   - Overview
   - $DLR_{ifd}$ syntax and semantics
   - $CM_{com}$

3 Comparison
   - ER and EER
   - UML class diagrams
   - ORM and ORM2

4 Discussion and Conclusions
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 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 DLRs 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 $DLR$ 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 $DLR$s are available now
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)
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 and look ahead to results

- DLR_{ifd} used to formally define the generic common conceptual data modeling language $CM_{com}$, i.e., with syntax and (model-theoretic) semantics.
- This $CM_{com}$ 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

- $DLR_{ifd}$ used to formally define the generic common conceptual data modeling language $CM_{com}$, i.e., with syntax and (model-theoretic) semantics.
- This $CM_{com}$ 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 $\geq 2$) and arbitrary concepts according to the syntax:

$$
\begin{align*}
R &\rightarrow T_n \mid P \mid (i/n : C) \mid \neg R \mid R_1 \cap R_2 \\
C &\rightarrow T_1 \mid A \mid \neg C \mid C_1 \cap C_2 \mid \exists[i]R \mid \leq k[i]R
\end{align*}
$$

$i$ denotes a component of a relation; if components are not named, then integer numbers between 1 and $n_{\text{max}}$ are used, where $n$ is the arity of the relation. Only relations of the same arity can be combined to form expressions of type $R_1 \cap R_2$, and $i \leq n$.
The base language: **DLR**

The **model-theoretic semantics** of **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 \( R \) a subset \( R^{\mathcal{I}} \) of \( (\Delta^{\mathcal{I}})^n \), such that the conditions are satisfied following:

\[
\begin{align*}
\top^{\mathcal{I}}_n & \subseteq (\Delta^{\mathcal{I}})^n \\
\mathbb{P}^{\mathcal{I}} & \subseteq \top^{\mathcal{I}}_n \\
(-R)^{\mathcal{I}} & = \top^{\mathcal{I}}_n \setminus R^{\mathcal{I}} \\
A^{\mathcal{I}} & \subseteq \Delta^{\mathcal{I}} \\
\top^{\mathcal{I}}_1 & = \Delta^{\mathcal{I}} \\
(R_1 \cap R_2)^{\mathcal{I}} & = R_1^{\mathcal{I}} \cap R_2^{\mathcal{I}} \\
(-C)^{\mathcal{I}} & = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C_1 \cap C_2)^{\mathcal{I}} & = C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}} \\
(i/n : C)^{\mathcal{I}} & = \{(d_1, \ldots, d_n) \in \top^{\mathcal{I}}_n | d_i \in C^{\mathcal{I}}\} \\
(\exists[i] R)^{\mathcal{I}} & = \{d \in \Delta^{\mathcal{I}} | \exists(d_1, \ldots, d_n) \in R^{\mathcal{I}}. d_i = d\} \\
(\leq k[i] R)^{\mathcal{I}} & = \{d \in \Delta^{\mathcal{I}} | \{(d_1, \ldots, d_n) \in R^{\mathcal{I}}_1 | d_i = d\} \leq k\}
\end{align*}
\]
A knowledge base is a finite set $KB$ of $DLR$ (or $DLR_{ifd}$) axioms of the form $C_1 \subseteq C_2$ and $R_1 \subseteq R_2$.

An interpretation $\mathcal{I}$ satisfies $C_1 \subseteq C_2$ ($R_1 \subseteq R_2$) if and only if the interpretation of $C_1$ ($R_1$) is included in the interpretation of $C_2$ ($R_2$), i.e. $C_{\mathcal{I}(t)}^1 \subseteq C_{\mathcal{I}(t)}^2$ ($R_{\mathcal{I}(t)}^1 \subseteq R_{\mathcal{I}(t)}^2$).

$\top_1$ 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 DLRs

- ORM to ORM2
- ORM to UML
- ORM to EER
- ORM to ER

- UML to ORM2
- UML to DLR

- EER to ORM2
- EER to DLR

- ER to ORM2
- ER to DLR

Relationship between "fragments of ORM2" w.r.t. the common CDM languages

Extensions to DLR
- ORM2
- DLR
- ORM
- UML
- EER
- ER

Existing formal partial transformations between CDM languages

Existing diagram-based partial transformations between CDM languages

C. Maria Keet
A comparison of conceptual data modeling languages
\( \mathcal{DLR}_{ifd} \) has two additional constructs compared to \( \mathcal{DLR} \):

- Identification assertions on a concept \( C \), which has the form \((\text{id } C[i_1]R_1, \ldots, [i_h]R_h)\), where each \( R_j \) is a relation and each \( i_j \) denotes one component of \( R_j \).

- Non-unary functional dependency assertions on a relation \( R \), which has the form \((\text{fd } R i_1, \ldots, i_h \rightarrow j)\), where \( h \geq 2 \), and \( i_1, \ldots, i_h, j \) denote components of \( R \).

- Syntax and semantics as for \( \mathcal{DLR} \)
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 $\Sigma$.

- $\text{REL}$ 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 $R$.

- $\ldots$
Example: syntax for $CM_{com}$

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

Equivalent representation in $DLR_{ifd}$ as: Author $\sqsubseteq$ Person (subsumption), Author $\sqsubseteq \exists[\text{auth}]\text{writes}$ (at least one), Author $\sqsubseteq \neg\text{Editor}$ (disjoint), Person $\sqsubseteq$ Author $\sqcup$ Editor (covering), and Person $\sqsubseteq \exists^{-1}[\text{From}]\text{id}$, $\top \sqsubseteq \exists^{\leq 1}[\text{To}]\,(\text{id} \sqcap \exists[\text{From}] : \text{Person})$ (key)
Example: syntax for $CM_{com}$

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

Equivalent representation in $DLR_{id}$ as: $\text{Author} \sqsubseteq \text{Person}$ (subsumption), $\text{Author} \sqsubseteq \exists[auth]\text{writes}$ (at least one), $\text{Author} \sqsubseteq \neg\text{Editor}$ (disjoint), $\text{Person} \sqsubseteq \text{Author} \sqcup \text{Editor}$ (covering), and $\text{Person} \sqsubseteq \exists^{=1}[\text{From}]\text{id}$, $\top \sqsubseteq \exists^{\leq1}[\text{To}](\text{id} \sqcap [\text{From}] : \text{Person})$ (key)
For each Person, exactly one of the following holds:
   some Author is that Person; some Editor is that Person.
It is possible that more than one Author writes the same
   Book and that the same Author writes more than one Book.
Each Book, Author combination occurs at most once in the
   population of Author writes Book.
Each Author writes some Book.
   For each Book, some Author writes that Book.

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

Let $\Sigma$ be a $CM_{com}$ conceptual data model. An interpretation for the conceptual model $\Sigma$ is a tuple $B = (\Delta^B \cup \Delta^B_D, \cdot^B)$, such that:

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

A comparison of conceptual data modeling languages
Definition ($CM_{com}$ Semantics cont'd)

$B$ 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 $C_1, C_2 \in C$: if $C_1$ ISA$_C C_2$, then $C_1^B \subseteq C_2^B$.
- For each $R_1, R_2 \in \mathcal{R}$: if $R_1$ ISA$_R R_2$, then $R_1^B \subseteq R_2^B$.
- For each $U_1, U_2 \in \mathcal{U}$, $R_1, R_2 \in \mathcal{R}$,
  \[
  \text{REL}(R_1) = \langle U_1 : o_1, \ldots, U_n : o_n \rangle,
  \text{REL}(R_2) = \langle U_2 : o_2, \ldots, U_m : o_m \rangle, \quad n = m, \quad R_1 \neq R_2: \text{ if } U_1 \text{ ISA}_U U_2, \text{ then } U_1^B \subseteq U_2^B.
  \]
Definition \((C\text{M}_{\text{com}} \text{ Semantics cont'd})\)

For each \(C \in \mathcal{C}, R_h \in \mathcal{R}, h \geq 1,\)

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

\[
\begin{align*}
o_a &= t_1[U_1] = \ldots = t_h[U_h] \\
o_b &= s_1[U_1] = \ldots = s_h[U_h]
\end{align*}
\]

\[
\begin{align*}
t_j[U] &= s_j[U], & j \in \{1, \ldots, h\}, \text{ and for } U \neq j
\end{align*}
\]

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 \(j \in \{1, \ldots, h\}\), and \(o_b\) is an instance of \(C\) that is the \(U_j\)-th component of a tuple \(s_j\) of \(R_j\), for \(j \in \{1, \ldots, h\}\), and for each \(j\), \(t_j\) agrees with \(s_j\) in all components different from \(U_j\), ...
Definition ($\mathcal{CM}_{\text{com}}$ 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 $i \geq 2$, $i \neq j$,
  \[
  \text{REL}(R) = \langle U_1 : C_1, \ldots, U_i : C_i, j : C_j \rangle,
  \]
  \[
  \text{FD}(R) = \langle U_1, \ldots, U_i \to j \rangle, \]
  then for all $t, s \in R^B$, we have that $t[U_1] = s[U_1], \ldots, t[U_i] = s[U_i]$ 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
  \[
  \text{REL}(R_i) = \langle U_i : C_i, \ldots U_m : C_m \rangle \]
  (and, thus, $R_i \in R_i^B$ and $o_j \in C_j^B$), if \{U_1, U_2, \ldots U_{i-1}\} REX U_i, then
  \[
  \forall i \in \{1, \ldots, i\}. o_j \in C_j^B \rightarrow \text{CMIN}(o_j, r_i, u_i) \leq 1 \land u_i \neq u_1 \land \ldots \land u_i \neq u_{i-1} \text{ where } u_i \in U_i^B, r_i \in R_i^B. \]
Overview

Relationship between "fragments of ORM2" w.r.t. the common CDM languages

Extensions to \( \text{DLR} \)

Existing formal partial transformations between CDM languages

Existing diagram-based partial transformations between CDM languages

C. Maria Keet A comparison of conceptual data modeling 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 $\text{CARD}$ is restricted to any of the values $\{\geq 0, \leq 1, \geq 1\}$, denoted in $\Sigma$ with $\text{CARD}^-$.

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 \((\mathcal{CM}_{\text{UML}})\)

A \(\mathcal{CM}_{\text{UML}}\) 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 \(\mathcal{CM}_{\text{com}}\) syntax and semantics, except for the aggregation association \(\text{PW}\), with syntax

\[
\text{PW} = \langle U_1 : C_1, U_2 : C_2 \rangle,
\]

that has no defined semantics.
Definition \( (CM_{ORM}) \)

A \( CM_{ORM} \) conceptual data model is a tuple
\[
\Sigma = (L, REL, ATT, CARD, ISA_C, ISA_R, ISA_U, KEY, EXTK, FD, OBJ, REX, RDM, JOIN, KROL, \text{RING}^-)
\]
adhering to \( CM_{com} \) syntax and semantics, and, in addition, such that:

- **JOIN** comprises the following constraints: \{join-subset, join-equality, join-exclusion\} over \( \geq 2 \) \( n \)-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].
A $CM_{ORM2}$ conceptual data model is a tuple 
$\Sigma = (L, REL, ATT, CARD, ISA_C, ISA_R, ISA_U, DISJ, COVER, KEY, EXTK, FD, OBJ, REX, RDM, JOIN, KROL, RING)$ 
adhering to the syntax and semantics as defined for $CM_{com}$, 
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], $CM_{UML}$ not a proper fragment of $CM_{ORM}$ (total exclusive subtypes–but OCL).
  - $CM_{UML}$ fragment of $CM_{ORM2}$ (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 $CM_{com}$...
Discussion

- Comparison trivial (almost) with the 5 definitions
- Four finer-grained issues
  - With ORM formalization of [H89], $CM_{UML}$ not a proper fragment of $CM_{ORM}$ (total exclusive subtypes–but OCL).
  - $CM_{UML}$ fragment of $CM_{ORM2}$ (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 $CM_{com...}$
Discussion

Why a comparison with $DLR_{ifd}$ and $CM_{com}$ 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.
  - $CM_{UML}$, $CM_{ER}$, and $CM_{EER}$ are in ExpTime-complete ($DLR_{ifd}$ 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]
## Discussion and Conclusions

### Features [KR07]

<table>
<thead>
<tr>
<th>Language ⇒ Feature \downarrow</th>
<th>Lite</th>
<th>OWL DL</th>
<th>v1.1</th>
<th>DL-Lite</th>
<th>( F )</th>
<th>( R )</th>
<th>( A )</th>
<th>ifd</th>
<th>( \mu )</th>
<th>reg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role hierarchy</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>N-ary roles (where ( n \geq 2 ))</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>±</td>
<td>±</td>
<td>±</td>
<td>±</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Role concatenation</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Role acyclicity</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Symmetry</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Role values</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Qualified number restrictions</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>One-of, enumerated classes</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Functional dependency</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Covering constraint over concepts</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Complement of concepts</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Complement of roles</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Concept identification</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Range typing</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Reflexivity *</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Antisymmetry *</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Transitivity * †</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Asymmetry †</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>±</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Irreflexivity ‡</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>

C. Maria Keet

A comparison of conceptual data modeling languages
## Features [KR07]

<table>
<thead>
<tr>
<th>Language ⇒ Feature</th>
<th>Lite</th>
<th>OWL DL</th>
<th>v1.1</th>
<th>DL-Lite</th>
<th>ifd</th>
<th>µ</th>
<th>reg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role hierarchy</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>N-ary roles (where $n \geq 2$)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>±</td>
<td>±</td>
<td>±</td>
<td>+</td>
</tr>
<tr>
<td>Role concatenation</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Role acyclicity</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Symmetry</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Role values</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Qualified number restrictions</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>One-of, enumerated classes</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Functional dependency</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Covering constraint over concepts</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Complement of concepts</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Complement of roles</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Concept identification</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Range typing</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Reflexivity *</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Antisymmetry *</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Transitivity * ‡</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Asymmetry ‡</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>±</td>
<td>-</td>
</tr>
<tr>
<td>Irreflexivity ‡</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>
Given a knowledge base $\mathcal{K} = (T, R, A, F)$ 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 $R_b \in R$;
- $R_b$ does not occur in any identification assertion, i.e., for $(\text{id} C[i_1]R_1, \ldots, [i_h]R_h)$ then $R_b \neq R_1, \ldots, R_b \neq R_h$. 
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 $DLR_{ifd}$ and $CM_{com}$ 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
  - $CM_{UML}$, $CM_{ER}$, and $CM_{EER}$ are in ExpTime-complete ($DLR_{ifd}$ is)
  - Knowledge about computationally more appealing fragments in NP or NLogSpace [ACKRZ07, KS06, SCS07]
- $DLR_{ifd}$ 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
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 $CM_{com}$ simplifies (semi-)automated interoperability of conceptual data models in different graphical languages

- “$DLR_{\mu ifd}$” for ORM’s ring constraints, temporal extensions with $DLR_{US}$ and $ER_{VT}$
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 $CM_{com}$ simplifies (semi-)automated interoperability of conceptual data models in different graphical languages

- “$DLR_{\mu,ifd}$” for ORM’s ring constraints, temporal extensions with $DLR_{US}$ and $ER_{VT}$
The DLR family

Comparison

Discussion and Conclusions


Background
The DLR family
Comparison
Discussion and Conclusions


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