Measuring Expressiveness in Conceptual Modeling

Susanne Patig

Otto-von-Guericke-University of Magdeburg, PO Box 4120
D-39016 Magdeburg, Germany
`susanne.patig@iti.cs.uni-magdeburg.de`

Abstract. Expressiveness refers to things said in a description or sayable in a description language. It contributes to the quality of conceptual schemas and justifies the development of new modeling languages. Different notions of expressiveness exist, originating from different fields of computer science. In this paper, a framework is developed that defines different types of expressiveness, integrating the other notions. The framework contains a cardinal measure to assess the expressiveness of both descriptions and description languages.

1 Introduction: The Importance of Expressiveness

Conceptual modeling means describing reality with the help of some *description language* (*modeling language*) to create a more or less formalized *description* (*conceptual model* or *schema*). Typically, this process is characterized by offering many degrees of freedom and requiring a large number of subjective decisions. First of all, depending on the purpose of conceptual modeling, a description language must be chosen. In addition to the purpose, this choice is influenced by the preferences of the modeler. Furthermore, even if the same description language is used to model a given part of reality for the same purpose, often, alternative descriptions can be created, reflecting, for instance, subjectivity in recognizing reality or in applying the description language. Although each of the alternative descriptions may be correct, it may possess a different quality, that can influence information system design and implementation.

One of the properties that contribute to the quality of a description is expressiveness [2]. As a provisional definition, *expressiveness* (synonyms: modeling power, expressive power) refers to things said in a description or sayable in a description language. It is widely assumed that the more expressive a description language is, the more expressive its descriptions will be. Altogether, a description language must be "sufficiently expressive" for a given purpose, that is, a description language that cannot express all things required should not be chosen [20].

Apart from being an aspect of quality, expressiveness becomes important in transforming descriptions ([5], [18]). A transformation between two descriptions should preserve their expressiveness, i.e., the transformed description should be as expressive as the original one. Finally, "missing expressiveness" is often the reason to modify existing description languages [25].

In spite of its importance, the problem of expressiveness in conceptual modeling has hardly been addressed ([10], [11], [28]). Furthermore, different notions of expres-

A. Persson and J. Stirna (Eds.): CAiSE 2004, LNCS 3084, pp. 127–141, 2004.

siveness exist, originating from different fields of computer science (see section 2). It is the aim of this paper to develop a general framework that gives an explicit characterization of expressiveness and contains other notions of expressiveness as special cases (see section 3). Moreover, we propose a cardinal measure of expressiveness that is applicable when other approaches fail (see section 3). The paper concludes by applying the framework to evaluate the expressiveness of descriptions and description languages within entity-relationship modeling (see section 4).

2 Previous Research: Expressiveness – Meaning and Measurement

Many authors assume an “intuitive” notion of expressiveness, i.e., they use the term without a definition. Their usage of the term implies that an evaluation of expressiveness is always based on some reference. Three types of *references* can be identified:

- **[Reference R1]** Modeling a *universe of discourse (UoD)* always serves a certain purpose. In conceptual modeling, the purpose either requires one to represent the UoD as completely as possible or to extract special aspects from it, e.g. dimensions and facts in multidimensional data modeling. Expressiveness is greatest when it is possible to say all the things necessary to meet a particular purpose [17], [25]. Hence, the *purpose of conceptual modeling* serves as reference to evaluate expressiveness.
- **[Reference R2]** In the field of database query languages [7] or constraint programming languages [15] expressiveness is evaluated with reference to the *set of all things sayable*, e.g., the set of all queries that can be formulated. The more that can be said absolutely, the greater expressiveness is.
- **[Reference R3]** Expressiveness is evaluated with reference to the *things said elsewhere*, i.e., in another description [5] or description language. This type of reference is employed to assess the expressiveness of planning formalisms [23], logics [18] or database schemas [19], [21]. Equivalent expressiveness is given if all the things said elsewhere can be formulated. *Schema equivalence*, which is discussed in the context of database design, establishes a special case of equivalent expressiveness.

Measuring expressiveness is typically based on sets or on mappings. Using *sets*, a description language is at least as expressive as another one:

- **[Approach S1]** if the set of its symbols contains the symbols of the other description language (implicit [25]) or
- **[Approach S2]** if the statements representable by the description language contain all the statements that can be formulated by the other one ([18], [26])

as a proper subset. In order to form sets that share elements, the symbols or statements must be translated into a common description language.

To measure expressiveness by *mappings*, every element of a certain set is assigned one and only one element of another set. In detail, mappings are established between:

- **[Approach M1]** a philosophical ontology and the symbols of a description language ([10], [11]),
- **[Approach M2]** the sets of symbols belonging to different description languages [17],
- **[Approach M3]** the sets of statements of different descriptions [1], [3], [23], [21], [19].

If a mapping is found the sets are at least equally expressive. Mappings directly translate the elements of a set into the elements of the other one.

All approaches to measure expressiveness are confronted with the purely *syntactic part* of a (description) language, which consists of symbols and their legitimate connections, and the *semantic part*, which relates syntactic expressions to structures beyond language. Semantics describes the "meaning" of syntactic expressions. By *statement* we mean a syntactic expression and its meaning.

Usually in evaluating expressiveness, syntactically different statements must be compared. Their equivalent expressiveness can be shown by syntactic transformations (schema transformation, e.g. [21],[13]). Even if such transformations can be found it must be ensured that the meaning of the statements is preserved, hence, evaluating expressiveness relates to semantics. The approaches S1, M1 and M2 to do not look at semantics in a formalized way, whereas the approaches S2 and M3 mostly use ideas from logic.

Mathematical logic provides us with means to deal with the duality of syntax and semantics. Using logics as description languages, different *descriptions*, i.e., different sets of logical formulas, are equally expressive if they have the same (formal) model [1], [21]. A *model* denotes the formal semantics of a logical description, i.e., an assignment of values to variables (*interpretation*) such that all formulas are true.

A model can be understood as a set of elements from a UoD whose properties satisfy the formulas of the logical description. From the *extensional point of view*, two models are equal if they comprise the same elements. Using first-order predicate logic to give an example, two descriptions composed of the atomic formulas `smallest_even_number(X)` and `smallest_prime_number(Y)` respectively, have the same model consisting in the natural number two (*example 1*). So, it is not possible to discriminate between both descriptions by extension. However, in line with the *intensional point of view*, the two descriptions are distinguishable by the separate characteristics they refer to [22].

We start building our framework for evaluating expressiveness from the extensional point of view. But, as example 1 indicates, the extensional point of view does not take into consideration special characteristics that must be described. Therefore, the framework also includes a cardinal measure that assesses expressiveness from the intensional point of view.

3 A Framework for Evaluating Expressiveness

The proposed framework is applicable to evaluate the expressiveness of either descriptions or description languages. To derive the framework, we assume that the intensional semantics of the descriptions, or of the symbols of the description languages, is representable by sets of logical formulas, for which (formal) models can be determined.

For the sake of simplicity, we use the term *description* in the following to refer to both descriptions and description languages. Hence, D_1 and D_2 denote separate descriptions, whose expressiveness is to be compared. The (sets of) models of the descriptions are represented by $M(D_1)$ and $M(D_2)$, respectively. Throughout this work we concentrate on minimal models. An interpretation I is called a *minimal model* of a description D if it is a model of D and if no other interpretation I' exists such that $I' \subset I$ and I' is a model of D as well [4]. The notion of minimal models is equivalent to the closed world assumption [24] because it declares that all relevant statements are contained in the (minimal) model.

Since models correspond to sets of elements, we can compare the expressiveness of different descriptions from the extensional point of view by analyzing the relationship between these sets. Any two sets are either equal or disjoint, or one set is a proper subset of the other one or they intersect. We use these types of relationships to define expressiveness from the extensional point of view (see table 1):

- **[Definition DF1]** Two descriptions D_1 and D_2 are *extensionally equally expressive* if every model of D_1 is also a model of D_2 and vice versa: $M(D_1) = M(D_2)$. This corresponds to the theoretical foundation of approach M3.
- **[Definition DF2]** A description D_1 is *extensionally more expressive* than a description D_2 *in the sense of generality* if all models of D_1 are proper supersets of the models of D_2: $M(D_1) \supset M(D_2)$. So, *generality* means an increase of extension. In other words, a larger number of elements satisfy all formulas of the description. Since only minimal models are considered, adding new formulas to descriptions is a necessary, but not a sufficient, condition for increasing expressiveness in the sense of generality. This point is addressed later on.
- **[Definition DF3]** A description D_1 is *extensionally more expressive* than a description D_2 *in the sense of precision* if all models of D_1 form proper subsets of the models of D_2: $M(D_1) \subset M(D_2)$. Hence, *precision* corresponds to a decrease in extension, i.e., there exists a smaller number of elements such that all formulas of the description have value “true”. As regards precision, adding new formulas to descriptions is neither necessary nor sufficient for increasing expressiveness.

In the following cases measuring expressiveness from the extensional point of view is impossible:

- **[Definition DF4]** Two descriptions D_1 and D_2 are *comparable* concerning expressiveness if their models intersect: $M(D_1) \cap M(D_2) \neq \emptyset$.
- **[Definition DF5]** If all models of two descriptions D_1 and D_2 are disjoint, i.e., $M(D_1) \cap M(D_2) = \emptyset$, their expressiveness *cannot be compared*, since the statements of the descriptions refer to different sets of elements.

If semantics is considered, the approaches presented in section 2 mainly concentrate on equivalent expressiveness. Superior expressiveness is only proven if a description (language) subsumes another description (language) [26], [1], [3], [21]. Neither approach distinguishes between generality and precision.

The above definitions directly support two types of references for evaluating expressiveness: Definition DF1 complies with reference R3, i.e., the things said elsewhere, whereas definitions DF2 and DF3 are compatible with reference R2, which relates greater expressiveness to the possibility of saying more. To this point, reference R1, the purpose of conceptual modeling, has not been integrated into the framework.

The framework for evaluating expressiveness presented so far can be applied independent of how the models are determined. But, under certain circumstances, determining models can be computationally hard or even impossible [22]. Furthermore, focusing all attention on the models of descriptions neglects the intensional point of view and does not yield expressiveness results in case of intersecting models.

To combine the extensional and the intensional point of view within our framework (see table 1) and to include the purpose of modeling, we define **[Definition DF6]** that the *expressiveness* is greater the more that can be said for a certain purpose. The purpose of conceptual modeling is formalized with the help of a reference description RD. This allows one to derive a measure E(RD, CD) to evaluate the expressiveness of a description CD (*description of comparison*) relative to the reference description.

By *reference description* we mean a set of statements that depend on the purpose of conceptual modeling and are representable by logical formulas. The *extension* of the reference description consists of the set of elements that are a model of every formula. To comply with the extensional definitions of expressiveness given above, the reference description must satisfy the following *requirements*:

- **[Requirement RQ1]** A *first-order predicate logic* [22] is used because of the comprehensible definition of its models. In order to simplify the specification, we agree on the following:
    - The set of elements is decomposed into several distinct domains, called *sorts,* to distinguish variables by type. For each sort, we assume that we have a *binary predicate "=" for equality*, which is interpreted as equality in all models.
    - All variables are universally quantified.
    - Any variables that are not explicitly equal must be distinct.
- **[Requirement RQ2]** The set of formulas must be *consistent*, since otherwise the reference description does not posses a model [22].
- **[Requirement RQ3]** The formulas that form the reference description must be *independent* of each other, i.e., each formula is neither synonymous with, nor a logical consequence of, other formulas of the reference description. *Synonymy* is given if formulas have the same intension; this is decided by the purpose of conceptual modeling. To recall example 1 from section 2, the formulas `smallest_even_number(X)` and `smallest_prime_number(Y)` are not synonymous since the differ in intension, whereas `smallest_prime_number(Y)` and `smallest_number_above_one_and_divisible_by_one_and_itself(Z)` are. Logical consequence is defined in the usual sense [22]. Independence is required by the measure E(RD, CD) proposed. Furthermore, adding dependent formulas to a reference description changes neither its extension nor its intension and hence does not affect expressiveness. Adding an independent formula to a reference description D_1 (see table 2) alternatively:

- increases the extension of the reference description, which is equivalent to increasing expressiveness in the sense of generality from the extensional point of view (definition DF2 and description D_2, table 2),
- decreases its extension, leading to an increasing expressiveness in the sense of precision (definition DF3 and description D_3, table 2),
- leaves its extension unchanged, as in the case in example 1 from section 2.

Table 1. Extensional expressiveness – an example

	Reference Description		
	D_1	**D_2**	**D_3**

Logical Formulas	`green(X)`	`green(X)`
		`has_thorns(Y)`
			`green(X)`
			`has_thorns(Y)`
			`= (X, Y)`

$M(D_i)$ Real-World Semantics	all green individuals	all green individuals, all thorny individuals, all individuals that are both thorny and green	all individuals that are both thorny and green
Extensional Expressiveness		$M(D_1) \subset M(D_2)$	$M(D_1) \supset M(D_3)$
		more expressive than D_1 in the sense of	
		generality	precision

To summarize, requiring that the reference description must consist of independent formulas results in *minimal reference descriptions* for a given purpose, i.e., eliminating any of its formulas changes the intension of the reference description and thus affects the purpose.

– **[Requirement RQ4]** The reference description must consist of *literals*, i.e., atomic formulas or their negation [22]. *Atomic formulas* are constructed by combining predicate symbols and terms. Constant symbols, variable symbols or function symbols that are applied to terms constitute *terms* [22]. Literals are required because each literal represents one and only one *intension* (a characteristic or a relationship) and an extension, so its contribution to expressiveness can be easily identified. In contrast, the complex formula `green(X)` $\wedge$ `cactus(X)` $\Rightarrow$ `has_ thorns(X)` combines different intensions, which would be difficult to measure if only some characteristics were contained in the description whose expressiveness is evaluated.

After the reference description RD has been created, it can be compared with other descriptions CD to evaluate their expressiveness. For the description CD it must be indicated (based on the intension) whether a certain statement of the reference description is contained (1) or not (0). Table 3 summarizes the possible combinations of occurring statements; their quantities are denoted by s, t, u and v. Because minimal models are considered, statements must be contained in the reference description or in the description to be compared ($v = 0$, $s + t + u > 0$).

Table 2. Comparing descriptions

		Reference Description RD		**Sum**
		statement **contained**	statement **not contained**	
Description of Comparison CD	statement **contained**	s	t	s + t
	statement **not contained**	u	v = 0	u
Sum		s + u	t	s + t + u

The *cardinal measure E(RD, CD)* determines the degree of expressiveness depending on the similarity between the reference description RD and the description CD to be compared. It is derived from measuring similarity between binary vectors, e.g., by means of the Tanimoto, the Simple Matching or the Rassel-Rao coefficient [16]. Taking into account v = 0, these coefficients are calculated as follows:

$$E(RD, CD) = \frac{s}{s+t+u} \tag{1}$$

Expressiveness E(RD, CD) can take values between zero and one (maximum expressiveness). The expressiveness of the description that constitutes the reference description (i.e., E(RD, RD)) always amounts to one.

The measure E(RD, CD) reflects expressiveness from the intensional point of view without being in contradiction with the extensional point of view: If conceptual modeling aims at generality, then the description that is extensionally the most expressive in the sense of generality must be chosen as reference description. In that case, the reference description contains *all* statements (t = 0), some might be missing in the description CD ($u \geq 0$).

Furthermore, it is possible to bias E(RD, CD) towards extensional expressiveness in the sense of precision by choosing the extensionally most precise description as reference description. Because of the extensional definition of precision (definition DF3), some statements might be lacking in the reference description ($t \geq 0$). Although this seems to be somewhat counter-intuitive, remember that we require of the reference description to be minimal for a given purpose. If we want to describe only all green individuals (description D_1, table 2), any description that says more than this is less expressive, because it either weakens this statement by including more individuals than required for the purpose (description D_2 in table 2) or makes it stronger by excluding individuals that are needed for the purpose (description D_3 in table 2).

If the extensional definition of expressiveness is not appropriate because the models of two descriptions intersect (definition DF4), the intensional measure E(RD, CD) can be applied, choosing one of the descriptions as reference description. Alternatively, a special reference description can be created that contains all statements required for a certain purpose (t = 0). A special reference description is also able to reveal differences in the intensional expressiveness of descriptions that are extensionally equally expressive (definition DF1).

Finally, by using a special reference description, the other approaches to measure expressiveness can be integrated into our framework. In this case, the statements of the reference description refer either to a philosophical ontology (approach M1) or to the symbols of a description language (approaches S1, M2). The requirements a reference description must satisfy will help to identify unnecessary or redundant symbols ("syntactic sugar", see section 4). Furthermore, our framework provides these approaches with a semantic foundation, which has been missing so far. Table 1 summarizes the proposed framework for measuring expressiveness.

Table 3. A framework for evaluating expressiveness

	The expressiveness of two descriptions D_1 and D_2 is				
	equal	**greater** in the sense of		**comparable**	**not comparable**
		generality	**precision**		
		in favor of D_1			
EX	$M(D_1) = M(D_2)$	$M(D_1) \supset M(D_2)$	$M(D_1) \subset M(D_2)$	$M(D_1) \cap M(D_2) \neq \emptyset$	$M(D_1) \cap M(D_2) = \emptyset$
IN	$E(RD, CD) = \frac{s}{s+t+u} \quad s+t+u > 0$				—
	RD = purpose $t = 0, u \geq 0$	RD = D_1 $t = 0, u \geq 0$	RD = D_1 $t \geq 0, u \geq 0$	RD = D_1 or RD = D_2 $t > 0, u > 0$ alternatively: RD = purpose $t = 0, u \geq 0$	

Abbreviations: D: description, M: model, EX: extensional, IN: intensional

4 Applying the Framework to Measure Expressiveness

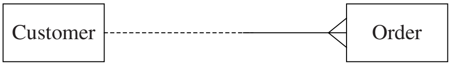
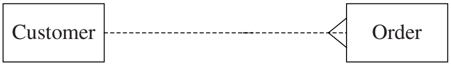
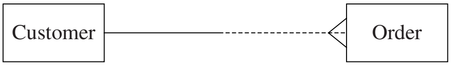
The first example analyzes the expressiveness of different descriptions. The descriptions are given as *entity-relationship (ER)* diagrams, see table 4. Following the Chen notation [8], the ER diagram D_1 in table 4 does not show whether customers without orders may exist. On the contrary, in the ER diagrams D_2 to D_5 dotted lines indicate that customers without orders or orders without customers are allowed [12]. Taking ER diagram D_2 as an example, every order *must* be assigned *one and only* one customer, whereas each customer *may* place *several* orders.

Within our framework, evaluating expressiveness either from the extensional point of view or by applying the measure E(RD, CD) is based on consistent sets of statements (table 4). Unlike in [13], we do not aim at finding transformations between the sets of statements assigned to each ER diagram. Rather, we are in line with [6], which analyze similarities between descriptors associated with schemas. In contrast to both approaches we do not derive the statements automatically from the descriptions[1] but create them manually so that they observe the requirements of the reference description.

---

[1] This would be a mainly syntactic view and restrict the approach to a certain description language.

Table 4. The expressiveness of different descriptions

	Entity Relationship Diagrams	**Sets of Statements**	**Extensional Expressiveness**
D_1		$=(\text{ordered}(o), c)$	--
D_2		$=(\text{ordered}(o), c)$ $\text{ordering}(\hat{c}, \hat{o})$ $\neg(\text{ordering}(\bar{c}, \bar{o}))$	$M(D_1) \subset M(D_2)$
D_3		$=(\text{ordered}(o), c)$ $\text{ordering}(\hat{c}, \hat{o})$	$M(D_1) = M(D_3)$
D_4		$=(\text{ordered}(o), c)$ $\neg(=(\text{ordered}(\tilde{o}), \tilde{c}))$ $\text{ordering}(\hat{c}, \hat{o})$ $\neg(\text{ordering}(\bar{c}, \bar{o}))$	$M(D_1) \subset M(D_4)$ $M(D_2) \subset M(D_4)$ $M(D_3) \subset M(D_4)$ $M(D_5) \subset M(D_4)$
D_5		$=(\text{ordered}(o), c)$ $\neg(=(\text{ordered}(\tilde{o}), \tilde{c}))$ $\text{ordering}(\hat{c}, \hat{o})$	$M(D_1) \subset M(D_5)$

In creating the sets of atomic and independent formulas of a sorted first-order predicate logic for each ER diagram (see table 4), we use the sorts `order` $(o, \hat{o}, \bar{o}, \tilde{o})$ and `customer` $(c, \hat{c}, \bar{c}, \tilde{c})$, the variables of which are enclosed in brackets. The operation `ordered` is a mapping that assigns one and only one element of the sort `customer` to each element of the sort `order`. Furthermore, the predicate `ordering` establishes a relationship between a customer and an order.

Assessing expressiveness from the extensional point of view means analyzing the models of the sets of statements associated with the ER diagrams. The ER diagram D_1 is completely represented by the single formula `=(ordered(o), c)`, because the mathematical definition of a mapping allows both customers with several orders and customers without orders, but this is not visible from the ER diagram. However, since we concentrate on minimal models, customers without orders do not belong to the models $M(D_1)$. Therefore, the models of the ER diagrams D_1 and D_3 are identical, i.e., both descriptions are extensionally equally expressive, see table 4.

Because minimal models are considered, statements on non-existence must be included in the reference description and thus stated explicitly (similarly in [20]). So, the extensions of the descriptions increase and the models $M(D_1)$ are included as proper subsets if customers without orders (ER diagrams D_2, D_4) or orders without customers (ER diagrams D_4, D_5) are permissible. Adding `ordering(`$\hat{c}$`,` $\hat{o}$`)` to a set of statements does not change its extension but introduces a new, independent intension: From the fact that each order is assigned a customer we cannot conclude that each customer has placed an order.

From the extensional point of view, ER diagram D_4 is the most expressive one in the sense of generality, whereas the ER diagrams D_1 and D_3 are extensionally equally expressive and the most expressive ones in the sense of precision. Hence, the measure

E(RD, CD) must be applied to reveal a difference in the expressiveness of the ER diagrams D_1 and D_3. If the purpose of conceptual modeling consists in creating the most precise description, the set of statements associated with ER diagram D_3 must form the reference description, as it makes it clear that neither orders without customers nor customers without orders are allowed. Since only one of the statements of the reference description D_3 is associated with ER diagram D_1, the measure E(RD, CD) = E(D_3, D_1) amounts to 0,5 (1/2), i.e., ER diagram D_3 is more expressive than ER diagram D_1 concerning precision (table 5). In measuring expressiveness in the sense of generality, the statements assigned to ER diagram D_4 form the reference description.

Table 5. Expressiveness E(RD, CD) of the ER diagrams from table 5

	Reference description RD representing		
Description to be compared CD	*precision*	*generality*	*arbitrary purpose*
	D_3	D_4	D_2
D_1	1/2 = 0,5	1/4 = 0,25	1/3 = 0,33
D_2	2/3 = 0,67	3/4 = 0,75	**3/3 = 1**
D_3	**2/2 = 1**	2/4 = 0,5	2/3 = 0,67
D_4	2/4 = 0,5	**4/4 = 1**	3/4 = 0,75
D_5	2/3 = 0,67	3/4 = 0,75	2/4 = 0,5

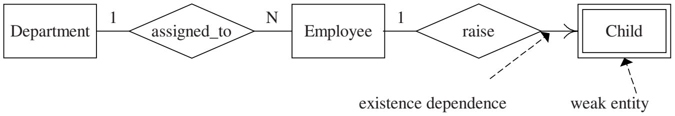
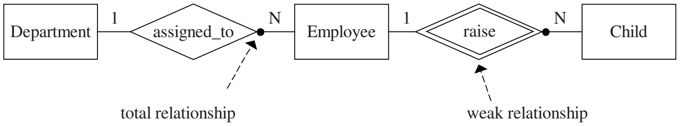
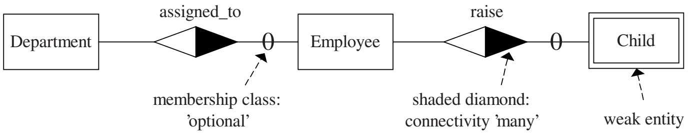
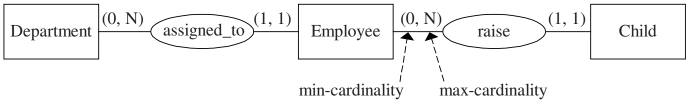
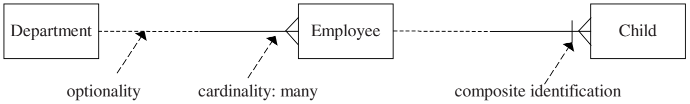
Looking at the first example, the Chen notation seems to be less expressive than other ER notations. This is examined in more detail in the second example, which treats different ER notations as different description languages. We restrict ourselves to constructs related to relationship types, since they vary strongly in terminology and symbolism. Because of their historical importance and their specific appearance, the notations proposed by Chen [8], Scheuermann et al. [27], Teorey et al. [29], ISO [14] and Barker [12] are chosen, see table 6.

In table 6, the ER notations are illustrated by a simple example: A department may have many employees, although (e.g., newly established) departments without any employee may exist. Each employee must be assigned to one and only one department. Employees may have none or several children. Each child belongs to one and only one employee. The identifier of each child contains the identifier of the employee the child belongs to.

The ER notations offer a different number of symbols to express this example. Furthermore, the names of the symbols vary, see table 6. This raises the question how many, and which, symbols are necessary to describe relationship types in general. In other words, it must be examined how expressive the five ER notations are for this purpose.

Using the proposed framework to answer this question, a reference description must be created. In this case, the statements of the reference descriptions refer to the things concerning relationship types that must be represented by the symbols of a description language. Table 7 summarizes the statements of the reference description, which will be explained in the following. To keep the logical formulas simple, the variables `e` and `r`, standing for the sorts `entity_type` and `relationship_type` respectively, are individualized by the symbol ‘!’. This means that ‘!e’ and ‘!r’ occurring in different logical formulas symbolize different variables.

Table 6. Different ER notations

Entity-Relationship Notations by Example	**Symbols provided**
C: Chen (1976) [8]	
	1) lines 2) diamond 3) number 4) existence dependence 5) weak entity
S: Scheuermann et al. (1980) [27]	
	1) lines 2) diamond 3) number 4) total relationship 5) weak relationship
T: Teorey et al. (1986) [29]	
	1) lines 2) shaded diamond 3) membership class 4) weak entity
I: ISO (1987) [14]	
	1) lines 2) oval 3) min-cardinality 4) max-cardinality
B: Barker (1990) [12]	
	1) lines 2) optionality 3) cardinality 4) composite identification

For the purpose of describing relationship types, the following things (intensions) must be sayable by a description language:

- Every relationship type has a *degree* [29] that specifies the number of entity types that are associated with the relationship type. This corresponds to the operation `degree: relationship_type` → `integer`. The statements in table 7 represent independent special cases in which the degree is 2 (binary, statement 1), 1 (unary, statement 2) or larger than 2 (n-ary, statement 3), respectively.
- All instances of an entity type **must** participate one time (`min_participate(!e, !r, 1)`, statement 4) or a certain number of times (`min_participate(!e, !r, N)`, statement 5) in the formation of instances of the relationship type. The intension of statement 4 is described by the terms *min-cardinality* [14], *total relation-*

Table 7. Expressiveness of the ER notations in describing relationship types

		Description of Comparison									
Reference Description		**C**		**S**		**T**		**I**		**B**	
Statements	**RD**	**CD**	**Sy**	**CD**	**Sy**	**CD**	**Sy**	**CD**	**Sy**	**CD**	**Sy**
1) = (degree(!r), 2)	1	1	1, 2	1	1, 2	1	1, 2	1	1, 2	1	1
2) = (degree(!r), 1)	1	1	1, 2	1	1, 2	1	1, 2	1	1, 2	1	1
3) > (degree(!r), 2)	1	1	1, 2	1	1, 2	1	1, 2	1	1, 2	0	
4) min_participate(!e, !r, 1)	1	0	4?	1	4	1	1	1	3	1	1
5) min_participate(!e, !r, N)	1	0		0		0		1	3	0	
6) max_participate(!e, !r, 1)	1	1	3	1	3	1	2	1	4	1	3
7) max_participate(!e, !r, N)	1	1	3	1	3	1	2	1	4	1	3
8) option_participate(!e, !r)	1	0		1	1	1	3	1	3	1	2
9) id_dependence(!e, !r)	1	1	4, 5	0	5?	1	4	0		1	4
E(RD, CD), RD: statements 1-9	1	6/9		7/9		8/9		8/9		7/9	
E(RD, CD), RD: statements 1-8	1	5/8		7/8		7/8		8/8		6/8	

Abbreviations: RD: binary vector of the reference description, CD: binary vector of the description of comparison, Sy: Symbols

ship [27], *mandatory membership class* [29] or simply *mandatory* [12]. In the logical formulas 4-7, the constant symbols ‘1’ (representing ‘one’) and ‘N’ (representing ‘above one’) are used directly.

- All instances of an entity type participate in at most one instance of the relationship type (`max_participate(!e, !r, 1)`, statement 6) or in more than one instance (`max_participate(!e, !r, N)`, statement 7), respectively. These intensions are represented alternatively by *numbers* ([8], [27]), *max-cardinalities* [14], *shaded diamonds* [29] or *cardinalities* [12].
- It is possible that some instances of an entity type are not involved in instances of the relationship type (`option_participate(!e, !r)`, statement 8). In the ER notations this is called *optional* [12] *membership class* [29] or represented by the *min-cardinality* ‘0’ [14]. Finally, the intension of statement 8 is assumed for all relationships that are not total [27].
- The identifiers (primary keys) of the instances of a weak entity type are composed, including as a part the identifiers of those entity instances to which the weak entity instances are related (`id_dependence(!e, !r)`, statement 9). Apart from *weak entity* ([8], [29]), this intension is named *existence dependence* [8], *weak relationship* [27] or *composite identification* [12].

Table 7 also contains the binary vector of the reference description. The binary vectors of the ER notations follow straightforwardly from the explanations above; the symbols (the numbers refer to table 6) that lead to marking a statement as contained (‘1’) are given. Concerning the Chen notation, the symbol for existence dependence does not seem to correspond to the intension of statement 4 because in another paper [9] a new symbol for this intension is added. Moreover, weak relationships only partly comply with the intension of statement 9, since “... [composite identification of the weak entity instances; *the author*] ... this need not always be the case” [27, p. 125].

By applying the measure E(RD, CD), it becomes obvious (see table 6) that expressiveness is not dependent on the number of special symbols a description language provides: Though it offers five symbols, the Chen notation is less expressive than the ISO notation, which uses only four symbols. This is due to the fact that the symbols 4 and 5 in the Chen notation redundantly express the intension of statement 9.

Furthermore, if we excluded statement 9 from the reference description, because the composition of identifiers is important only for database implementation and not for conceptual modeling, which deals with describing the UoD, the ISO notation would be the most expressive one.

5 Conclusion and Future Research

We have presented a framework for evaluating expressiveness in conceptual modeling. Expressiveness is discussed in different fields of computer science using different terms. The proposed framework shows that the different discussions can be integrated to define and measure expressiveness in a general and consistent manner.

In contrast to approaches concerning schema equivalence, our framework is applicable not only to descriptions but also to description languages. In both cases expressiveness may be assessed from the extensional point of view or from the intensional one, using the cardinal measure E(RD, CD). To apply the measure E(RD, CD), a reference description RD must be constructed, which must satisfy certain requirements. On the one hand, these requirements join the two different semantic views (extensional, intensional) on expressiveness. On the other hand, the requirements become important in assessing description languages. Requirement RQ3 especially, reveals that the expressiveness of a description language increases only if orthogonal symbols are added to a minimal set of orthogonal symbols. In addition, the reference description may be used as a general guideline for comparing different description languages with reference to a certain purpose, leading to criteria of comparison that are constructed in a systematic way determined by the requirements RQ1-RQ4.

As the second example indicates, in many cases there will be different description languages that are sufficiently and equally expressive for a certain purpose. Hence, choosing among description language does not depend only on their expressiveness but also on additional criteria. Such additional criteria may be how easy it is to say things by using the description language or how easily the things said can be understood. Few approaches for measuring these additional criteria have been presented so far [20], so this is a topic for future research.

Finally, the cardinal value of expressiveness we obtain by applying the measure E(RD, CD) helps to gain insight into the evolution of description languages. For instance (see the second example in section 4), the expressiveness of the first ER notation (Chen (1976)) was increased by adding symbols for the mandatory (statement 4 in table 7) or optional (statement 8 in table 7) participation of entity instances in the instances of an relationship type. Actually, the second example is a part of larger research work, which aims at discovering factors that influence the development of description languages in general. This research will be continued.

References

1. Baader, F.: A Formal Definition for the Expressive Power of Terminological Knowledge Representation Languages. Journal of Logic and Computation 6 (1996) 33-54
2. Batini, C., Ceri, S., Navathe, S.B.: Conceptual Database Design. 4th ed. Benjamin Cummings, Redwood City et al. (1992)
3. Borgida, A.: On the relative expressiveness of description logics and predicate logics. Artificial Intelligence 82 (1996) 353-367
4. Braß, S.: Vervollständigungen für Logikdatenbanken. Forschungsbericht 315. Informatik IV, Universität Dortmund (1980)
5. Cali, A., Calavese, D., De Giacomo, G., Lenzerini, M.: On the Expressive Power of Data Integration Systems. In: Spaccapietra, S., March, T.S., Kambayashi, Y. (eds.): Conceptual Modeling – ER 2002. LNCS, Vol. 2503. Springer, Berlin et al. (2002) 338-350
6. Castano, S., De Antonellis, V., Fugini, M.G., Pernici, B.: Conceptual Schema Analysis: Techniques and Applications. ACM Transactions on Database Systems 23 (1998) 286-332
7. Chandra, A.K.: Theory of Database Queries. In: Proceedings of the 7th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. ACM Press, Austin/Texas (1988) 1-9
8. Chen, P.P.: The Entity-Relationship Model - Toward a Unified View of Data. ACM Transactions on Database Systems 1 (1976) 9-36
9. Dogac, A., Chen, P.P.: Entity-Relationship Model in the ANSI/SPARC Framework. In: Chen, P.P. (ed.): Entity-Relationship Approach to Information Modeling and Analysis. North-Holland, Amsterdam (1983) 357-374
10. Green, P., Rosemann, M.: Integrated Process Modeling: An Ontological Evaluation. Information Systems 25 (2000) 73-87
11. Gregersen, H., Jensen, C.S.: On the Ontological Expressiveness of Temporal Extensions to the Entity-Relationship Model. In: Chen, P.P. (ed.): Advances in Conceptual Modeling – ER '99. LNCS, Vol. 1727. Springer, Berlin et al. (1999) 110-121
12. Halpin, T.: Information Modeling and Relational Databases – From Conceptual Analysis to Logical Design. Morgan Kaufmann, San Francisco et al. (2001)
13. Halpin, T.A., Proper, H.A.: Database Schema Transformation & Optimization. In: Papazoglou, M.P. (ed.): OOER '95: Object-Oriented and Entity-Relationship Modeling. LNCS, Vol. 1021. Springer, Berlin et al. (1995) 191-203
14. ISO: Information processing systems - Concepts and terminology for the conceptual schema and the information base. Technical Report 9007. **Ort** (1987)
15. Jeavons, P., Cohen, D., Gyssens, M.: How to Determine the Expressive Power of Constraints. Constraints – An International Journal 4 (1999) 113-131
16. Joly, S., Le Calvé, G.: Similarity functions. In: Van Cutsem, B. (ed.): Classification and Dissimilarity Analysis. Springer, Berlin et al. (1994) 67-86
17. Kung, C.H.: An analysis of three conceptual models with time perspective. In: Olle, T.W., Sol, H.G., Tully, C.J. (eds.): Information Systems Design Methodologies: A Feature Analysis. North-Holland, Amsterdam (1983) 141-167
18. Kurtonina, N., de Rijke, M.: Expressiveness of concept expressions in first-order description logics. Artificial Intelligence 107 (1999) 303-333
19. Larson, J. A., Navathe, S. B., Elmasri, R.: A Theory of Attribute Equivalence in Databases with Application to Schema Integration. IEEE Transactions on Software Engineering 15 (1989) 449-463
20. MacKinlay, J., Genesereth, M.R.: Expressiveness and language choice. Data & Knowledge Engineering 1 (1985) 17-29
21. McBrien, P., Poulovassilis, A.: A Formalization of Semantic Schema Integration. Information Systems 23 (1998) 307-334
22. Monin, J.-F.: Understanding Formal Methods. Springer, London (2003)

23. Nebel, B.: Die Ausdrucksstärke von Planungsformalismen: Eine formale Charakterisierung. Künstliche Intelligenz 13 (1999) 12-19
24. Reiter, R.: On closed world databases. In: Gaillaire, H., Minker, J. (eds.): Logic and Data Bases. Plenum, New York (1978) 55-76
25. Sapia, C., Blaschka, M., Höfling, G., Dinter, B.: Extending the E/R Model for the Multidimensional Paradigm. In: Kambayashi, Y., Lee, D.L., Lim, E.-P., Mohania, M.K., Masunga, Y. (eds.): Advances in Database Technologies. LNCS, Vol. 1552. Springer, Berlin et al. (1999) 105-116
26. Schäuble, P., Wüthrich, B.: On the Expressive Power of Query Languages. Report 173. Department Informatik, Institut für Informationssysteme, ETH Zürich (1992)
27. Scheuermann, P., Schiffner, G., Weber, H.: Abstraction Capabilities and Invariant Properties Modelling within the Entity-Relationship Approach. In: Chen, P.P. (ed.): Entity-Relationship Approach to Systems Analysis and Design. North-Holland, Amsterdam (1980) 121-140
28. Si-Said Cherfi, S., Akoka, J., Comyn-Wattiau, I.: Conceptual Modeling Quality – From EER to UML Schemas Evolution. In: Spaccapietra, S., March, T.S., Kambayashi, Y. (eds.): Conceptual Modeling – ER 2002. LNCS, Vol. 2503. Springer, Berlin et al. (2002) 414-428
29. Teorey, T.J., Yang, D., Fry, J.P.: A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model. ACM Computing Surveys 18 (1986) 197-222