# Verbalizing ORM Models in Malay and Mandarin

Shin Huei Lim and Terry Halpin

INTI International University, Malaysia
shinhuei.lim@i.edu.my, t.halpin@live.com

**Abstract.** The rich graphical notations provided by fact-oriented modeling approaches such as Object-Role Modeling (ORM) for capturing business constraints assist modelers to visualize fine details of their data models. However, the data models themselves are best validated with domain experts by verbalizing the models in a controlled natural language, and by populating the relevant fact types with concrete examples. While a number of fact-based modeling tools provide extensive verbalization support in English, comparatively little work exists to provide fact-based model verbalization support for other languages, especially Asian languages. This paper describes our initial work on verbalizing ORM models in Bahasa Malaysia (Malay) and Mandarin. We discuss aspects of these languages that are not found in English, which require special treatment in order to render natural verbalization (e.g. noun classifiers, and the order in which sentence elements are placed), and describe our current implementation efforts, which involved creating both a prototype and an extension to the NORMA (Natural ORM Architect) tool.

## 1 Introduction

A conceptual data model includes a conceptual schema and a population of instances. The fact structures and business rules (constraints or derivation rules) are best validated with subject matter experts, who best understand the business domain. Since such domain experts often lack technical expertise in the graphical languages used by modelers to capture the model, model validation is best performed by verbalizing the model to the domain experts in a controlled natural language (an unambiguous subset of natural language with restricted grammar and vocabulary) and by populating the relevant fact types with concrete examples.

This process of validation by verbalization and population is central to, and facilitated by, fact-oriented modeling approaches, such as Object-Role Modeling (ORM) [11, 12, 16], Cognition enhanced Natural Information Analysis Method (CogNIAM) [23], and Fully Communication Oriented Information Modeling (FCO-IM) [2], which use fact types as their sole data structure, unlike attribute-based approaches such as Entity Relationship (ER) [4] modeling and the class diagramming technique in the Unified Modeling Language (UML) [24].

While some fact-based modeling tools provide extensive verbalization support in English, comparatively little work exists to provide fact-based model verbalization support for other languages, especially Asian languages. As far as we know, no work

Y.T. Demey and H. Panetto (Eds.): OTM 2013 Workshops, LNCS 8186, pp. 545–554, 2013.

has yet been published on verbalizing conceptual data models in Bahasa Malaysia (Malay) or Mandarin. This paper describes our initial work on verbalizing ORM models in Malay and Mandarin (more specifically, the Simplified Chinese version of Mandarin, as adopted in mainland China, Singapore and Malaysia). We discuss aspects of these languages that require special treatment in order to render natural verbalization (e.g. noun classifiers, and the order in which sentence elements are placed), and describe our current implementation efforts, which involved creating both a prototype and an extension to the NORMA (Natural ORM Architect) tool.

The rest of this paper is structured as follows. Section 2 briefly overviews related work on high level textual languages for data modeling, both within and outside the fact-oriented community. Section 3 provides a brief summary of how various logical constructs relevant to verbalization are mapped to Malay and Mandarin, and discusses our overall strategy for dealing with noun classifiers. Section 4 illustrates our prototype implementation for verbalizing ORM constraints in Malay and Mandarin. Section 5 summarizes the main contributions and outlines future research directions.

## 2 Related Research

To our knowledge, the first controlled natural language for conceptual data modeling was Reference and Idea Language (RIDL) [22], which was developed in the 1980s based on a binary relationship version of NIAM; the model declaration part was implemented in the RIDL* tool, but the query part was never implemented. Afterwards, other fact-oriented languages were developed. In the late 1980s and the 1990s, one of us specified the first version of Formal ORM Language (FORML) to capture ORM constraints in textual form, and provided patterns to automatically generate verbalizations of constraints in some early ORM tools (InfoDesigner, InfoModeler, VisioModeler, Microsoft Visio for Enterprise Architects). The Language for Information Structure and Access Descriptions (LISA-D), based on Predicator Set Model (PSM), was specified [19] by researchers at Radboud University, but was not fully implemented. ConQuer, an ORM-based language for conceptual queries, was implemented in the ActiveQuery tool, which generated SQL code from ConQuer queries [3].

More recently, FORML version 2 (FORML 2) was implemented in the NORMA tool [5], providing enhanced, automated verbalization of constraints and derivation rules in second generation ORM (ORM 2) [9] in English [6, 14, 15]. Preliminary work has also been made to extend FORML 2 for full model input and queries [17].The Constellation Query Language (CQL) [18] is under development in the Active Facts tool to provide modeling and conceptual query functionality for ORM. The DogmaModeler tool has been extended to verbalize basic ORM constraints in ten languages (English, Dutch, German, Italian, Spanish, Catalan, French, Lithuanian, Russian and Arabic) [20], but while the range of languages addressed is impressive the verbalizations generated are not always grammatically correct (e.g. Each Person must Has at least one Name. Each Person WorksFora Company must AffiliatedWith that Company.).

Outside the fact-oriented modeling community, several high level textual modeling and/or query languages have been developed. The Object Constraint Language (OCL) [29] is used to augment UML class models with rules that cannot be expressed graphically in UML. However, OCL's attribute-based nature leads to semantic instability, its rule contexts are restricted to classes, and OCL expressions are often too technical for business users to understand and hence validate. Some textual languages for ER have been proposed (e.g. see section 16.3 of [16]), but these are limited in scope, and are prone to semantic instability caused by an underlying attribute-based model.

Controlled natural languages are often linguistics-based, employing a formal, executable subset of natural language (typically English). Attempto Controlled English (ACE) [1] supports a wide range of natural statements and queries, relying on interpretation rules (e.g. **and** has priority over **or**) to enable its text to be unambiguously translated into discourse representation structures, a syntactic variant of first-order logic (FOL). John Sowa's Common Logic Controlled English (CLCE) [28] has the full expressibility of FOL, but its use of untyped variables tends to make its expressions look more mathematical than natural. Processable ENGlish (PENG) [25] uses a controlled lexicon of predefined function words as well as domain-specific content words that can be defined by the author on the fly. For further details on controlled natural languages, see [21, 25, 27].

## 3 Logical Elements and Noun Classifiers

To automate verbalization of ORM constraints, we utilize patterns based on an intermediate, logical form, which is transformed to a linguistic form suitable to the target natural language. A mapping of basic ORM constraints to unsorted first order logic is provided in [7]. ORM 2 extended ORM with several constraint types, and is best formalized in terms of sorted first order logic supplemented by modal operators to indicate the constraint modality. For our most recent formalization of ORM 2, see [13].

The logic form patterns include slots for various logical elements such as quantifiers and operators which have corresponding textual representations in natural languages. Table 1 lists the correspondences for all the *modal operators* relevant to our verbalizations. We use Kripke semantics and define a possible world as a state of the information model (not necessarily a possible state of the real world being modeled). Alethic constraints restrict the possible states or state transitions of fact populations, e.g. each person was born on at most one date. Deontic constraints are obligations that restrict the permitted states or state transitions of fact populations. For example, in a model that reflects a monogamous culture it is obligatory that each person has at most one wife (a deontic constraint that may be violated in practice) [10].

**Table 1.** Modal operators and their verbalizations in three natural languages

| *Modality* | *Symbol* | *English* | *Malay* | *Mandarin* |
|---|---|---|---|---|
| Alethic | □<br>◊<br>~◊ | it is necessary that<br>it is possible that<br>it is impossible that | Ia adalah perlu bagi<br>Ia adalah mungkin bagi<br>Ia adalah mustahil bagi | 是必要的<br>是可能的<br>是不可能的 |
| deontic | ***O***<br>***P***<br>***F*** | it is obligatory that<br>it is permitted that<br>it is forbidden that | Iaa dalah wajib bagi<br>Ia adalah dibenarkan bagi<br>Ia adalah dilarang bagi | 是强制性的<br>是允许的<br>是被禁止的 |

**Table 2.** Basic quantifiers and their verbalizations (here, Δ denotes a noun classifier)

| *Symbol* | *quantifier kind* | *English* | *Malay* | *Mandarin* |
|---|---|---|---|---|
| ∀ | universal | each, for each | setiap, bagi setiap | 每Δ |
| ∃ | existential | some, at least one | seseΔ, sekurang-kurangnya satuΔ | 某一Δ, 至少一Δ |
| $\exists^{0..1}$ | at most 1 | at most one | paling banyak satuΔ | 最多一Δ |
| $\exists^{1}$ | exactly 1 | exactly one | hanya satuΔ | 只有一Δ |
| $\exists^{2..}$ | more than 1 | more than one, at least two | lebih daripada satuΔ | 多过一Δ, 至少两Δ |

Table 2 lists the most common quantifiers and their verbalizations in three languages. Here, the symbol "Δ" denotes any appropriate *noun classifier* for the term being quantified. Unlike English, when quantifiers are used Malay and Mandarin typically require additional words to classify the kind of thing being counted. Table 3 provides a few examples in Malay and Mandarin. Note that Mandarin often has many choices of classifier for the same usage category. In Malay, 'orang' can be a noun phrase or a classifier; if an entity type is named 'Orang' (meaning Person), no noun classifier is used for it. In Malay, for value type names such as 'Nama' ('Name'), 'Nombor Akaun Bank' ('Bank Account Number') etc., no noun classifier is used.

A detailed list of noun classifiers for Malay may be accessed online from *http://mbsskl.edu.my/panitia_bm/files/2009/02/mengenal-kata.pdf*. For a list of noun classifiers for Mandarin, see *http://xh.5156edu.com/page/z7949m2560j18586.html*.

Mappings for other logical or linguistic elements such as Boolean operators (**and**, **or**, **not**respectively render as "**dan**", "**atau"** and **"bukan**" in Malay, and as "和" and "或" and "不" in Mandarin.) and pronouns are also needed (e.g. "**that**" and "**the same**" render as "**itu**" and "**yang sama**" in Malay, and as "那Δ" and "一样的" in Mandarin).

Quantifiers are used in verbalizing many kinds of ORM constraints, including internal and external uniqueness and frequency constraints. We have space here to discuss only internal uniqueness constraints on binary fact types, but this will suffice to convey our general approach to enable incorporation of noun classifiers into the verbalization process. For discussing logical forms, we focus on the *n*:1 cases shown in Figure 1, but the forms for 1:*n*, 1:1 and *m:n* cases may be dealt with similarly.

**Table 3.** A small sample of noun classifiers in Malay and Mandarin

| *Noun classifier* | *Usage* | *Examples* |
|---|---|---|
| batang | For long and thin things | pencils, rivers, teeth |
| buah | For large or box-shaped objects | cars, houses, books |
| ekor | For all kinds of animal | ants, horses, elephants |
| orang | For humans | teachers, nurses, doctors |
| 根、条、只、瓶、… | For long and thin things | pencils, bottles, trees |
| 只、尾、条、头 … | For animals | fish, cats, dogs, cows |
| 本、册 | For book-like objects | books, diaries, albums |
| 个、位、名、… | For humans | teachers, nurses, doctors |

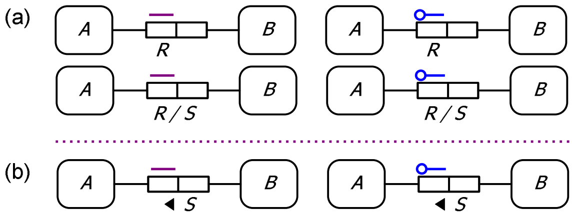


**Fig. 1.** Some uniqueness constraint cases for binary fact types

An ORM predicate may have many readings, each of which may be composed of various parts, such as front text and text parts (prebound, postbound, and trailing). The structure of a predicate reading is irrelevant to the logical form, which denotes a predicate by a single predicate symbol (e.g. *R* or *S*). Verbalizations may be displayed in positive form (e.g. **Each** Person was born on **at most one** Date.) or negative form (e.g. **It is impossible that some** Person was born on **more than one**Date.).

Table 4 shows the logical forms of the verbalizations for the cases in Figure 1, with positive (+ve) verbalizations shown first. For each case, the alethic constraint is shown before the deontic constraint. The Malay and Mandarin forms include a noun classifier (denoted here by $\Delta$) to categorize the kind of thing being counted. In Malay, the modal operator appears first (as in English), but in Mandarin it appears last. For positive verbalizations, the modal operator may be omitted (this is a user choice).

The absence of a simple, alethic uniqueness constraint on a role of an *n*:1 binary is also explicitly verbalized for the fact type (e.g. **It is possible that more than one** Person was born in **the same** Country). Such verbalizations translate in a similar way. For example the English logical form of the above verbalization is $\Diamond\exists y{:}B\exists^{2..}x{:}AxRy$. In Malay this becomes $\Diamond\exists y{:}B\exists^{2..}\Delta x{:}AxRy$; in Mandarin we get $\exists y{:}B\exists^{2..}\Delta x{:}AxRy\Diamond$.

**Table 4.** Logical forms of the constraint verbalizations for the cases in Figure 1

| *Case* | *English* | *Malay (omit* $\Delta$ *if type is a value type or is Orang)* | *Mandarin* |
|---|---|---|---|
| +ve (a) | $\Box\forall x{:}A\exists^{0..1}y{:}B\ xRy$<br>$\boldsymbol{O}\forall x{:}A\exists^{0..1}y{:}B\ xRy$ | $\Box\forall x{:}A\exists^{0..1}\Delta y{:}B\ xRy$<br>$\boldsymbol{O}\forall x{:}A\exists^{0..1}\Delta y{:}B\ xRy$ | $\forall\Delta x{:}A\exists^{0..1}\Delta y{:}B\ xRy\Box$<br>$\forall\Delta x{:}A\exists^{0..1}\Delta y{:}B\ xRy\boldsymbol{O}$ |
| +ve (b) | $\Box\forall x{:}A\exists^{0..1}y{:}B\ ySx$<br>$\boldsymbol{O}\forall x{:}A\exists^{0..1}y{:}B\ ySx$ | $\Box\forall x{:}A\exists^{0..1}\Delta y{:}B\ ySx$<br>$\boldsymbol{O}\forall x{:}A\exists^{0..1}\Delta y{:}B\ ySx$ | $\forall\Delta x{:}A\exists^{0..1}\Delta y{:}B\ ySx\Box$<br>$\forall\Delta x{:}A\exists^{0..1}\Delta y{:}B\ ySx\boldsymbol{O}$ |
| -ve (a) | $\sim\Diamond\exists x{:}A\exists^{2..}y{:}B\ xRy$<br>$\boldsymbol{F}\exists x{:}A\exists^{2..}y{:}B\ xRy$ | $\sim\Diamond\exists\Delta x{:}A\exists^{2..}\Delta y{:}B\ xRy$<br>$\boldsymbol{F}\exists\Delta x{:}A\exists^{2..}\Delta y{:}B\ xRy$ | $\exists\Delta x{:}A\exists^{2..}\Delta y{:}B\ xRy\sim\Diamond$<br>$\exists\Delta x{:}A\exists^{2..}\Delta y{:}B\ xRy\boldsymbol{F}$ |
| -ve (b) | $\sim\Diamond\exists x{:}A\exists^{2..}y{:}B\ ySx$<br>$\boldsymbol{F}\exists x{:}A\exists^{2..}y{:}B\ ySx$ | $\sim\Diamond\exists\Delta x{:}A\exists^{2..}\Delta y{:}B\ ySx$<br>$\boldsymbol{F}\exists\Delta x{:}A\exists^{2..}\Delta y{:}B\ ySx$ | $\exists\Delta x{:}A\exists^{2..}\Delta y{:}B\ ySx\sim\Diamond$<br>$\exists\Delta x{:}A\exists^{2..}\Delta y{:}B\ ySx\boldsymbol{F}$ |

## 4 Implementing ORM Verbalization in Malay and Mandarin

As indicated in Table 3, Mandarin differs from Malay in allowing more than one choice of noun classifier for the same usage category, or noun type. For example, the combination of mandatory role and ≥ 2 frequency constraints on Fishmonger's role in the fact type Fishmonger sells FishKind, may be verbalized in English as "**Each** Fishmonger sells **more than one**FishKind.". In Malay, the classifier for FishKind is "jenis". Showing logical words in **bold** and classifiers in red, this verbalizes in Malay as: "**Setiap** PenjualIkan menjual **lebih daripada satu** jenis SpesiesIkan.". In Mandarin however, one might choose any of the classifiers shown for humans in Table 3, so anyof these verbalizations could be used: 每**个**鱼贩卖**多过一**种鱼类; 每**位**鱼贩卖**多过一**种鱼类; 每**名**鱼贩卖**多过一**种鱼类. Which one is best, is decided by the user.

In a given ORM model, each object type has a distinct name. This name may be a simple noun but in general is a noun phrase (e.g. "Student", "Postgraduate Student", "Doctor or Dentist"). Using "NounType" for usage category, and "Classifier" for noun classifier, the situation for Mandarin may be modeled as shown in Figure 2(a). The fact type NounType has Classifier may be prepopulated with known data. However, in general the fact type NounPhrase is of NounType needs to be populated by the user. One reason for this to avoid massive databases (e.g. consider the number of noun phrases allowed for a give language). Another pragmatic reason is that the modelers often invent their own noun phrases to name various object types in their model, even though the terms they choose are not recognized in a standard language dictionary (e.g. "PostgradStudent", "PGstudent", or "Postgrad").

So the user interface for Malay or Mandarin needs to accept the noun type choice for a given object type from the user. Using the data in the prepopulated fact type for NounType has Classifier, the system can derive the allowed classifer(s), using the derivation rule shown in Figure 2. For Malay, where the NounType has Classifier is *n*:1, that derived classifier will be the only possibility. For Mandarin however, it will often be the case that more than one classifier is derived, so the user needs to be presented with the list of possible classifiers from which to choose his/her preferred one. As we illustrate later in this section, these requirements are met by our prototype tool.

The metamodel fragment in Figure 2(b) provides one way to view the situation if, instead of using a separate model for each language, one wishes to use a single model with multiple display options based on the language choice.

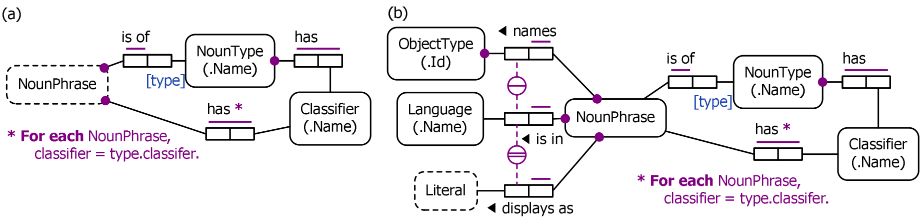


**Fig. 2.** Metamodel fragment for noun classifiers in (a) one language, (b) multiple languages

When implementing verbalization support for Malay, we initially utilized the verbalization framework built-in to the NORMA tool. NORMA's fact editor readily accepts fact types entered in any language using Roman characters, so this was no problem for entering Malay. Moreover, it was an easy task to provide modified snippets to choose corresponding logical elements such as quantifiers, and place them in different positions. However, dealing with noun classifiers requires extending NORMA's property sheets (e.g. adding properties for noun type and noun classifier in the properties window for object types). Moreover, extra work is required to modify the fact editor for input of Mandarin. With that in mind, before making these extensions to NORMA, we prototyped verbalization support for noun classifiers by building our own tool in C# for this purpose. We now briefly illustrate this implementation.

Figure 3 is a screenshot from our prototype tool for entering and verbalizing binary fact types in ORM using Bahasa Malaysia (shown here as the option BM) or Mandarin. In this example, Mandarin is selected, and the user is entering the fact type Country is the birth country of Politicianin Pinyin, the official phonetic script for translating Mandarin into Roman characters. In the upper screenshot shown, the user has just entered the word "shi". The system displays a list of Mandarin characters with this sound, for the user to choose the one with the intended meaning (in this case "是" meaning "is").

The noun phrases that name object types are entered in square brackets, facilitating parsing of the fact type text which in general may use mixfix predicate readings. The object type names are colored red, with the predicate text in blue. The lower screenshot in Figure 3 shows the complete entry for the fact type: 国家是政治家的出生地. Notice that the predicate text which is infix in English becomes mixfix in Mandarin with the second object type name embedded within the predicate text.

Pressing Ctrl+Enter causes the tool to draw a diagram for the fact type in the upper left window. As usual, each object placeholder in the predicate text is denoted by an ellipsis ("…"). The user now double-clicks each object type shape to invoke the dialog window for choosing the relevant noun classifier for that object type.


**Fig. 3.** Entering a fact type in Mandarin

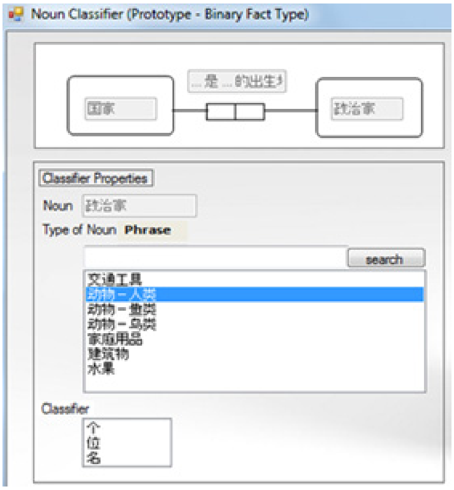


**Fig. 4.** Choosing the NounType and Noun Classifier

The name of the selected object type '国家'appears in the Noun box, and a list of noun types is displayed for the user to select the right one. As this list can be quite long, a search button enables the user to quickly find the correct type by entering its sound in Pinyin (this aspect is not shown here). In this case, the user selects the second item shown on the displayed list of noun types, and the relevant classifiers for this are displayed in the Classifier window. The user selects his/her preferred choice (in this case '个'). Similarly, for the other object type name, '政治家' the user chooses the relevant classifier (in this case '名'). The classifiers appear as tooltips when one mouses over the object type names.

Next the uniqueness pattern for the binary ($n$:1, $m$:$n$, 1:1, 1:$n$) is selected from a drop-down list. In this case, the user selects 1:$n$ (the same country may be the birth country of many politicians), and the tool displays the positive verbalization for the fact type, as shown in Figure 5. Object type names appear in purple, predicate text in green, logical elements (e.g. quantifiers) in blue, and noun classifiers in red.

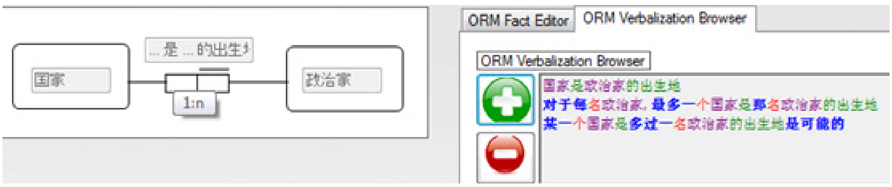


**Fig. 5.** Positive verbalization of a 1:$n$ fact type.

Table 5 shows the corresponding positive verbalizations in English and Mandarin. The negative form of the verbalizations may be displayed by selecting the negative (-) button. Verbalizations in Malay are performed in a similar manner. Figure 6 shows the final screen for the $n$:1 fact type Politician was born in Country.

**Table 5.** English and Mandarin verbalizations for fact type in Figure 5.

| *English* | *Mandarin* |
| --- | --- |
| Country is the birthcountry of Politician. | 国家是政治家的出生地 |
| For each Politician, at most one Country is the birthcountryof that Politician. | 对于每名政治家，最多一个国家是那名政治家的出生地 |
| It is possible that some Country is the birth-country of more than one Politician. | 某一个国家是多过一名政治家的出生地是可能的 |

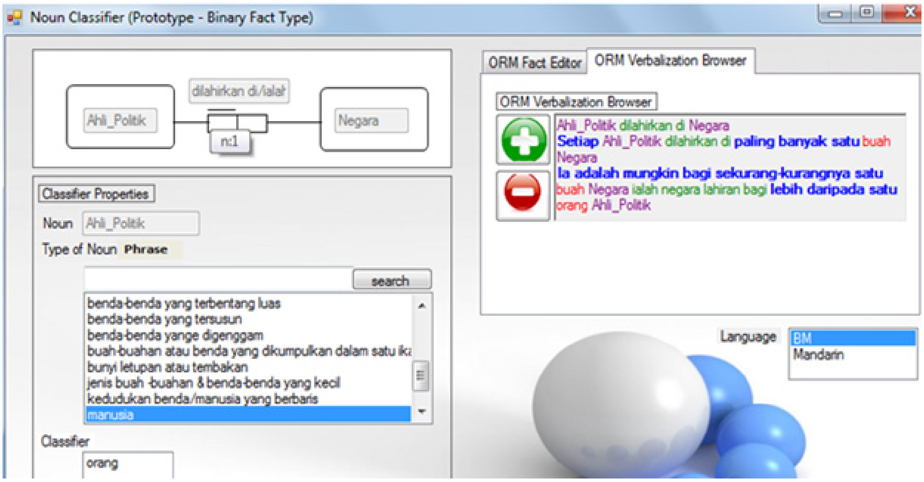


**Fig. 6.** Verbalization of an *n*:1 fact type in Malay

## 5 Conclusion

This paper described our initial work in verbalizing ORM models in Malay and Mandarin, with special attention to verbalizing noun classifiers. Future plans include implementing our approach via language extensions to the NORMA tool, and fully covering all of the many ORM graphical constraint varieties in these Asian languages.

## References


1. Attempto project (Attempto Controlled English), `http://attempto.ifi.uzh.ch/site/`
2. Bakema, G., Zwart, J., van der Lek, H.: Fully Communication Oriented Information Modelling. Ten Hagen Stam (2000)
3. Bloesch, A., Halpin, T.: Conceptual queries using ConQuer-II. In: Embley, D.W. (ed.) ER 1997. LNCS, vol. 1331, pp. 113–126. Springer, Heidelberg (1997)
4. Chen, P.P.: The entity-relationship model—towards a unified view of data. ACM Transactions on Database Systems 1(1), 9–36 (1976), `http://csc.lsu.edu/news/erd.pdf`
5. Curland, M., Halpin, T.: The NORMA Software Tool for ORM 2. In: Soffer, P., Proper, E. (eds.) CAiSE Forum 2010. LNBIP, vol. 72, pp. 190–204. Springer, Heidelberg (2011)

6. Curland, M., Halpin, T.: Enhanced Verbalization of ORM Models. In: Herrero, P., Panetto, H., Meersman, R., Dillon, T. (eds.) OTM-WS 2012. LNCS, vol. 7567, pp. 399–408. Springer, Heidelberg (2012)
7. Halpin, T.: A Logical Analysis of Information Systems: static aspects of the data-oriented perspective. Doctoral dissertation, University of Queensland (1989)
8. Halpin, T.: Business Rule Verbalization. In: Doroshenko, A., et al. (eds.) Proc. ISTA 2004. Lec. Notes in Informatics, vol. P-48, pp. 39–52 (2004)
9. Halpin, T.: ORM 2. In: Meersman, R., Tari, Z. (eds.) OTM-WS 2005. LNCS, vol. 3762, pp. 676–687. Springer, Heidelberg (2005)
10. Halpin, T.: Modality of Business Rules. In: Siau, K. (ed.) Research Issues in Sys. Analysis and Design, Databases and Software Dev., pp. 206–226. IGI Publishing, Hershey (2007)
11. Halpin, T.: Object-Role Modeling: Principles and Benefits. International Journal of Information Systems Modeling and Design 1(1), 32–54 (2010)
12. Halpin, T.: Fact-Orientation and Conceptual Logic. In: Proc. 15th International EDOC Conference, pp. 14–19. IEEE Computer Society, Helsinki (2011)
13. Halpin, T.: Formalization of ORM Revisited. In: Herrero, P., Panetto, H., Meersman, R., Dillon, T. (eds.) OTM-WS 2012. LNCS, vol. 7567, pp. 348–357. Springer, Heidelberg (2012)
14. Halpin, T., Curland, M.: Automated Verbalization for ORM 2. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1181–1190. Springer, Heidelberg (2006)
15. Halpin, T., Curland, M.: Enriched Support for Ring Constraints. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM-WS 2011. LNCS, vol. 7046, pp. 309–318. Springer, Heidelberg (2011)
16. Halpin, T., Morgan, T.: Information Modeling and Relational Databases, 2nd edn. Morgan Kaufmann, San Francisco (2008)
17. Halpin, T., Wijbenga, J.P.: FORML 2. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Ukor, R. (eds.) BPMDS 2010 and EMMSAD 2010. LNBIP, vol. 50, pp. 247–260. Springer, Heidelberg (2010)
18. Heath, C.: The Constellation Query Language. In: Meersman, R., Herrero, P., Dillon, T. (eds.) OTM 2009 Workshops. LNCS, vol. 5872, pp. 682–691. Springer, Heidelberg (2009)
19. ter Hofstede, A., Proper, H., van der Weide, T.: Formal definition of a conceptual language for the description and manipulation of information models. Information Systems 18(7), 489–523 (1993)
20. Jarrar, M., Keet, C.M., Dogilli, P.: Multilingual verbalization of ORM conceptual models and axiomatized ontologies. Technical Report, Vrije Universiteit Brussel, Brussels (2006)
21. Kuhn, T.: A Survey and Classification of Controlled Natural Languages, Computational Linguistics. MIT Press (to appear)
22. Meersman, R.: The RIDL Conceptual Language. Int. Centre for Information Analysis Services, Control Data Belgium (1982)
23. Nijssen, M., Lemmens, I.M.C.: Verbalization for Business Rules and Two Flavors of Verbalization for Fact Examples. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2008. LNCS, vol. 5333, pp. 760–769. Springer, Heidelberg (2008)
24. Object Management Group: OMG Unified Modeling Language (OMG UML), version 2.5 FTF Beta 1 (2012), `http://www.omg.org/spec/UML/2.5`
25. Pool, J.: Can Controlled Languages Scale to the Web? In: Proc. CLAW 2006 (2006), `http://utilika.org/pubs/etc/ambigcl/clweb.html`
26. Schwitter, R.: PENG (Processable English) (2007), `http://web.science.mq.edu.au/~rolfs/peng/`
27. Schwitter, R.: `http://sites.google.com/site/controllednaturallanguage/`
28. Sowa, J.: Common Logic Controlled English (2004), `http://www.jfsowa.com/clce/specs.htm`
29. Warmer, J., Kleppe, A.: The Object Constraint Language, 2nd edn. Addison-Wesley (2003)