Information Systems 60 (2016) 114–126

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Modeling dynamic relationship types for subsets of entity type instances and across entity types

P. Radha Krishna [a,*], Anushree Khandekar [b], Kamalakar Karlapalem [c]

[a] Infosys Limited, Manikonda Village, Gachibowli, Rangareddy District, Hyderabad 500032, Andhra Pradesh, India
[b] Microsoft R & D, Hyderabad, India
[c] International Institute of Information Technology, Hyderabad, India

ARTICLE INFO

Article history:
Received 5 July 2012
Received in revised form
10 March 2016
Accepted 14 March 2016
Available online 4 April 2016

Keywords:
ER model
Star constructs
Relational model
e-Contract meta-model

ABSTRACT

In a traditional ER model, once we specify a subclass or superclass relationship, any changes to that relationship are treated as schema evolution. Further, ER models are rigid in the sense that once a relationship type is specified across a set of entity types, an instance of relationship type occur when one instance of all participating entity types are specified. Therefore, it is difficult to introduce in a simplified manner all relationship types across subsets of given set of entity types. In this paper, we provide mechanisms to model in our extended ER model: (i) specification of dynamic relationship types across subsets of instances of entity types, (ii) a simplified specification of relationships across subsets of given set of entity types, and (iii) mapping our extended ER model to relational database schema. We also show through an e-contract example the utility of our extended ER model.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

ER model has constructs to capture the application requirements and conceptually represent them effectively. Conceptual modeling is a complex task that requires exploring various semantics present in the mini-world. In some scenarios, ER models do not allow or are not able to explicitly represent at the degree of precision required by the underlying concept for a particular application. For example, degree of precision: (i) fixes the participation and cardinality constraints ([1, n] is less precise than [2,8]), (ii) ability to model relationship among subsets of entities (addressed in this paper) is more precise than relationship type among entity types. Further, we also simplify the ER diagram by representing multiple relationship types among entity types by a simpler diagram. An ER model with large number of relationship types is difficult to comprehend and, for complex applications, expressiveness and clarity of conceptual model is also important.

Given an entity type and a set of entity instances, a subset of entities can be implicitly treated as a group; and this group of entities could be involved in relationships with other such groups from other entity types. There have been some approaches like, sub-classes [4,24] and categories [6] to handle some details of modeling relationships among subsets of entity types. These approaches dealt with the existence of a relationship between entity types at a higher level of abstraction. *Collection types*, presented in [12], are useful to model the relationship between collections of entity instances of participating entity types in a relationship. However, collection type does not capture additional application semantics such as specification of restrictions on number of entities in a collection (degree of precision is less). These intricate relationships among the groups of entities from different entity types need to be explicitly modeled at a substantial degree of precision. The category/collection types do not easily allow for multiple

* Corresponding author. Tel.: +91 40 67024610; fax: +91 40 23005223.
E-mail addresses: radhakrishna_p@infosys.com (P.R. Krishna), anushrik@microsoft.com (A. Khandekar), kamal@iiit.ac.in (K. Karlapalem).

http://dx.doi.org/10.1016/j.is.2016.03.010
0306-4379/© 2016 Elsevier Ltd. All rights reserved.

such groups to be generated dynamically and related to similar such groups. In fact, implementation of these aspects of conceptual model in implementation model is quite a challenge.

In order to categorize different entity types and their views, one has to dynamically modify ER model of an application for controlling the degree of precision to define sub-classes. For example, if there is a basketball player who is currently playing at club level, and in case he is also participating in the national team, we need to represent the 'Player' entity type with different degrees of precision using two different sub-classes. To support such additional application level requirements, we need to evolve new ER models. This gives rise to a complication of support for evolution, potentially after the database is populated. This is one of the reasons why current subclass mechanisms have not been used to cater to the kind of set relationship we address in this paper.

Further, in a traditional ER model, relationship type represents the relations between participating entity types. However, (i) this construct does not allow flexibility of participation of some of subsets of entity types in a relationship type, (ii) if there are different kinds of relationships between entity types, they need to be represented as separate relationship types, (iii) extending a relationship type by adding another entity type is treated as evolution of the ER model, *and is not considered as inherent property of a relationship type* (for quite a few applications one needs to add an entity type to existing relationship type or add a new relationship type among existing entity types), and (iv) there is no way of anticipating that subsets of given set of entity types can participate in multiple relationship types. The limitations mentioned above can be overcome by defining extensions to entity types and relationship types.

In this paper, we enhance the original ER model by extending entity type and relationship type constructs. These constructs balance between the expressiveness and complexity of the model while modeling complex and real-life database requirements. The main contributions of the paper are: (a) a new entity type named star-entity (*-entity) type which facilitates participation of subsets of entity instances is introduced; (b) a new relationship type named star-relationship (*-relationship) type which relaxes the relationship type to permit participation flexibility for entity types in a relationship type; (c) presented a case study for effective modeling of e-contracts database requirements by using new constructs and (d) the mapping of *-entity type and *-relationship type to relational data model is developed.

The rest of the paper is organized as follows. In Section 2, we present related work. Section 3 presents star (*)-entity and star (*)-relationship types and Section 4 presents an example on modeling e-contracts using star constructs. In Section 5, we discuss the mapping of new constructs into relational model and, in Section 6, we discuss about applicability of the proposed constructs. Section 7 concludes the paper.

2. Related Work

Chen [5] introduced the original ER model with the constructs entity type, relationship type and attributes. Entity types represent the things that can be uniquely identified and characterized by their attributes, whereas, relationship types represent associations among entity types. Attributes express information on entity and relationship types by mappings into value sets. The ER model also has general specializations of the basic constructs such as weak entity type, cardinalities constraints (one-to-one, one-to-many and many-to-many, existence-dependency) and keys [5,11,25]. Advances in the technology and complex applications necessitate extending the existent modeling capabilities to incorporate additional features. Several extensions to ER models are proposed in literature in order to represent complex applications. In [10], Geo-ER model is presented to conceptual model geographic applications. The Geo-ER model introduces new constructs to model spatial relationships and associated semantics. Temporal Entity-Relationship Model (TERM) is attempted in [16,17] to temporally extend the ER model by introducing time domain into conceptual modeling. Tauzovich [22] and Lai et al. [18] described extensions to ER model by incorporating temporal aspects. Gregersen and Jensen [9] presented a survey on temporally enhanced ER models.

In [14], composite entity type and table type relationship constructs are introduced for semantic modeling of databases in top down manner. In [21], an extension to ER model, known as ER-R model, to accommodate the concepts of events and rules is presented. The ER-R model combines event-based situation-action rules into the ER model for capturing application semantics and facilitates conceptual modeling of active databases. In [15], an ER^{EC} model has been presented to model *Electronic Contracts*. This model allows modeling exceptions and enables mapping conceptual model into workflows.

There are some works which focused on translating English like sentences into ER models. Hartmann and Link [13] presented conversion of application requirements specified in natural language sentences into ER model constructs, particularly Extended ER (EER) modeling features such as specialization, generalization and higher-order relationships. Bagui [3] described translation of XML schema into ER and EER models. Such translations are further helpful to capture more domain semantics into the conceptual models for complex applications.

Patig [19] presented a study on the evolution of ER models by considering over hundred ER models and categorized the ER evolution based on the Structure, Integrity, Behavior, Time, Uncertainty, Knowledge, Multi-dimensionality and Domain-specific. In [2], Badia stressed on the tradeoff between expressiveness and complexity, which must be considered while adding new constructs to the ER model. In [1,23], automated techniques for clustering related entities and relationships in large ER diagrams are presented for easy understanding and manage. Shovel et al. [20] presented a hierarchical entity-relationship diagram (HRED) that allows grouping of entities and relationships. HREDs facilitate abstraction of a complex ER diagram by a smaller abstract diagram;

Download English Version:

https://daneshyari.com/en/article/396787

Download Persian Version:

https://daneshyari.com/article/396787

Daneshyari.com