Automatic Generation of Extended ER Diagram Using Natural Language Processing

Dr. Muhammad Shahbaz [1], Dr. Syed Ahsan [2], Muhammad Shaheen [3], Rao Muhammad Adeel Nawab[4], Syed Athar Masood [5]
[1,2,3,4.] University of Engineering & Technology Lahore, Punjab Pakistan
[5]Department of Engineering Management, NUST College of E&ME, Rawalpindi Pakistan
[1] m.shahbaz@uet.edu.pk, [2] ahsancs@hotmail.com, [3] shaheen@uet.edu.pk, [5] atharmasood2000@hotmail.com

Abstract: Extended Entity Relationship Diagrams are an important step in information system design and software engineering. In the early seventies Peter Chen developed an efficient database management system, the ERD. Later on, ERD was enhanced to Extended ERD by adding new concepts like generalization and specialization.
The inspiration of EERD emerged from the common need to many organizations to have a unified methodology for file structure and database design. To meet the demands of users, to interpret problem statements in English, applying all the rules and generating an EERD. The structural approach is used to parse the sentences and tag them into different parts of the speech. The structural approach is used to map the tagged words into entities, attributes and relationships. [Dr. Muhammad Shahbaz, Dr. Syed Ahsan, Muhammad Shaheen, Rao Muhammad Adeel Nawab. Automatic Generation of Extended ER Diagram Using Natural Language Processing. Journal of American Science 2011;7(8)]. (ISSN: 1545-1003). http://www.americanscience.org.

Keywords: Geographic Information System (GIS), Usability, Interactivity, Human-GIS Interaction, Positional Accuracy, Hydrocarbon Exploration, Backpropogation Neural Network

1. Introduction

Databases today have become indispensable to almost any business carried out by an organization. So why not let Artificial Intelligence use its expert systems to handle the entire progression of construction - starting from a simple textual user input to the generation to EERDs (Extended Entity Relationship Diagram)? Application of structural analysis for the generation of EERD is something unprecedented in the history of Artificial Intelligence and Database Designing. Research along similar lines has been done previously but never ever has such a project been implemented.
During the stream of this project we have taken up the task of applying Structural Analysis to create the EERDs that could be further used to generate the tables in accordance with the normalization rules and keeping the functional dependencies intact. This would involve categorizing the parsed input as nouns, verbs and adjectives - a form that could be transformed and identified specifically as entities, relationships and attributes for the EERD. After the analysis and documentation phase we plan to implement the project along the following modules.

Module 1: Reading and parsing natural language input text given by the user.
Module 2: Heuristically classifying the text, that would serve as input to our next module.
Module 3: Generation of ERD and the final output in the form of a graphical diagram.
The third module is however mostly concerned with the generation of a text file that contains all the information needed to generate the ER diagram. This file would then be converted into a format that can be imported to an external tool. In our case the external tool is DeZign. In short the problem statement is very simple. Input in English language and the output is the desired ERD.

Relationship b/w Natural Language & Conceptual Modeling

Conceptual modeling has taken all its guidelines from artificial intelligence. The primitives in English and german language grammer are much more similar with ERD technique. [7].
[12] develops a dialogue tool with in a big project i.e. RADD (Rapid Application Development). The dialogue tool takes the input from the user in natural language, sample data is used to find out the semantic constraints on the database to be built. This work focuses on implementing the semantic constraints as it is the prerequisite for the normalization and denormalization or any other restructuring approach. Semantic constraints are important because they are necessary for the efficient and effective working of the database.

Differences between Data Modeling and Database Design

It is worth while to distinguish between Data Modeling and Database Design before discussing the various tools that are available in the market - on the internet - for the former.
Some of the most common Data Modeling techniques used today in the fields of Object Oriented Design (OOD) and Software Engineering

(SE) are the System Sequence Diagrams (SSDs), Data Flow Diagrams (DFDs), Use Case Diagrams (UCDs) and UML Diagrams, to name a few.

Entity Relationship Diagram (ERD) is just another representation of Data Modeling employed for designing databases. Various tools and software are available that assist in the drawing and diagrammatic illustration of ERDs. These tools are not responsible for automated generation of ERDs, but rather provide a platform for the user to graphically represent the information using various symbols. Some of these tools go as far ahead as to correct the mistakes that the user may have made in the course of drawing. Moreover, it is also possible to import the ERDs drawn further, to soft-wares that design the database from the respective ERDs. It should be noted here that there can be many representations of ERDs of the same problem statement. Similarly the database generated or designed can also vary from software to software for the same specific scenario. Hence one problem statement may lead to several different databases (i.e. different structures of databases) depending on the number of choices that one has in the intermediary steps.

Secondary Goal

The primary objective of this system is primarily defined in the overview statement i.e. the automated generation of an Extended Entity Relationship Diagram (EERD) through Structural Analysis. However secondary goals that could be achieved from this tool are as follows:

- The ERD generated would be exported to an external tool for further modification and correction.
- The backend tagging of the parsed words to different parts of speech could be used for similar purposes that require classification of words.

Implementation Details

In many information systems projects, requirements are primarily documented in English, and then database designers convert these English descriptions into database schemas in terms of ERDs. During the course of this project we have proposed a number of rules to generate an ERD diagram from English sentence structure. The basics constructs of English such as noun, verb, adjective, and adverb are found to have counterparts in the ERD. Finally and example is used to demonstrate the applicability of these rules in database design.

User Interface -Input Problem Statement

The program begins with the user interface as shown in Figure 2. User enters information and information is processed to generate an ERD using structural analysis approach. The front end input screen consists of six buttons each of which has a specific task which has been explained below.

Clear Text

When the user clicks on this button any previously written problem statement in the text field is erased. This is done so that the user can write a new problem statement.

Format Input

The user types the text in the input field shown as the white text field. The text entered by the user in the text field is formatted so that it can be send to the backend to be processed and finally generate an ERD. A space is put between full-stops and commas. Each sentence ends with a full stop.

Import:

The user has two options, he can either input problem statement directly into the text field by clicking on the button 'enter text', or he can import a previously stored problem statement from any directory in the computer.

When the user clicks on import the text field appears on the screen, this is where the imported file is displayed. To import a file it is essential for the user to give the destination of the file. One constraint to this is that the file has to be in either .txt or .rtf format to be imported.

Submit

Each sentence ends with a full stop. These sentences are sent to the back end one by one to be processed. The sentence is tagged using Brill's tagger, assigning each word to a particular part of speech. Using this information, rules are applied and relevant information consisting of which words are entities, attributes, relationships and cardinalities are stored in text files. These text files are then used to generate an ERD using the DeZign tool.

The submit button remains disabled when the user has not entered text, once the user types text into the text field this button is enabled and the user can now submit the problem statement for processing to generate an ERD.

Save:

The problem statement typed in the text field can be stored by the user for further use or reference using this option. The problem statement is saved as either a .txt or .rtf file in the directory in the computer whose location is given by the user.

Exit

To exit the program, the user clicks on this button and the program closes.

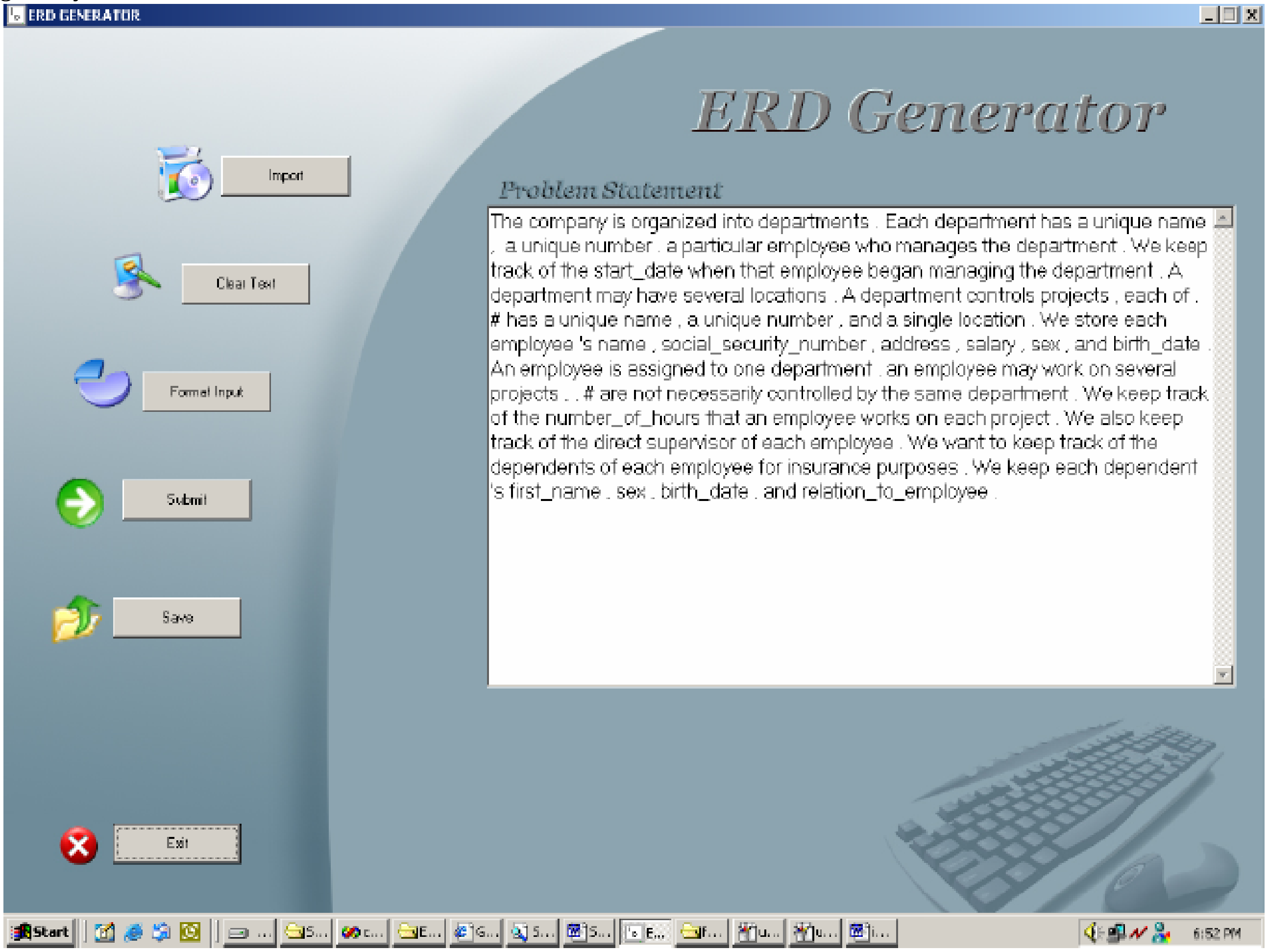

Figure 4.1 - User Interface

Input Assumptions/Constraints

Sentences are entered by the user either directly in the text field or the user has the option to import the problem statement from a text file. A number of limitations have to be put on the user when he types the description. These have been stated below:

1. User input should conform to all rules of English grammar.
2. It is recommended that user should input text in subject-verb-object format.
3. The software does not cater the first person. E.g. we assign a particular id to a department.
4. The words 'we', 'they', 'them', 'I', 'he', 'she' etc. are not allowed. First, second and third person is not allowed.
5. User cannot enter words like number of hours or phone number. These words should be of the form number_of_hours or phone_number. Words should be entered together as one noun.
6. Questions are not allowed in a scenario
7. Past tense not allowed.
8. Allow the word 'and' only when it terminates a list of attributes. E.g. the department consists of name, id and phone_number. Do not use 'and' otherwise in these sentences break the sentence into two.
9. Cannot use sentences like, e.g. usually each patient.

10. Cannot use semicolons, and other special characters. Commas and full stops are allowed. The system puts a space between the commas and full-stops on its own before running the rules algorithm.

In Brill a trainable rule based tagger is described that achieves performance comparable to that of stochastic taggers. Training this tagger is fully automated but unlike trainable stochastic taggers linguistic information is encoded directly in a set of simple non stochastic rules.
The primary goal of Eric Brill's research is to make information access and the use of computing devices a natural and painless task. As a step towards this goal, he is trying to make computers proficient at processing human language. He has pursued a line of research that falls under the rubric of Empirical Natural Language Processing [2, 3, 4, 5].

EERD Mapping
During the course of this project we have proposed some new rules for ERD generation from language constructs. The parts of speech in English language found its partners in ERD. An example is presented at the end.

Description of Rule

Some of the sample rules are given below. The list is huge and can be extended depending on the application and use.
Rule 1
A noun followed by a verb and then a noun forms: Both the nouns form the two entities
There exists one relationship, the verb, between the two entities.
English Statement: Various items are supplied by a supplier.
Analysis and translation: "Items" and "supplier" are nouns, they become the entity and "supplied" becomes the relationship between them. "Items" is changed to "item".
ERD: The corresponding ERD is shown in Figure 4.1.1.

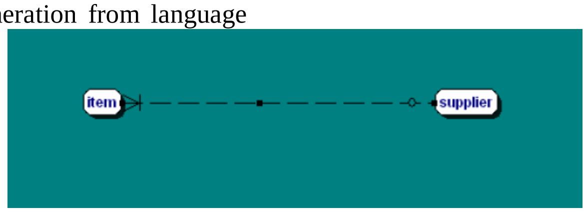

Figure. 4.1.1 - Rule 1

English Statement: Smith has a pen. Smith is from some oil company.

Analysis and translation: "Smith", "pen" and "company" are nouns which can be represented by entities in ERD. "has" and "is" are verbs for which relationships are facilitated in ERD.
ERD: The corresponding ERD is shown in Figure 4.1.2.

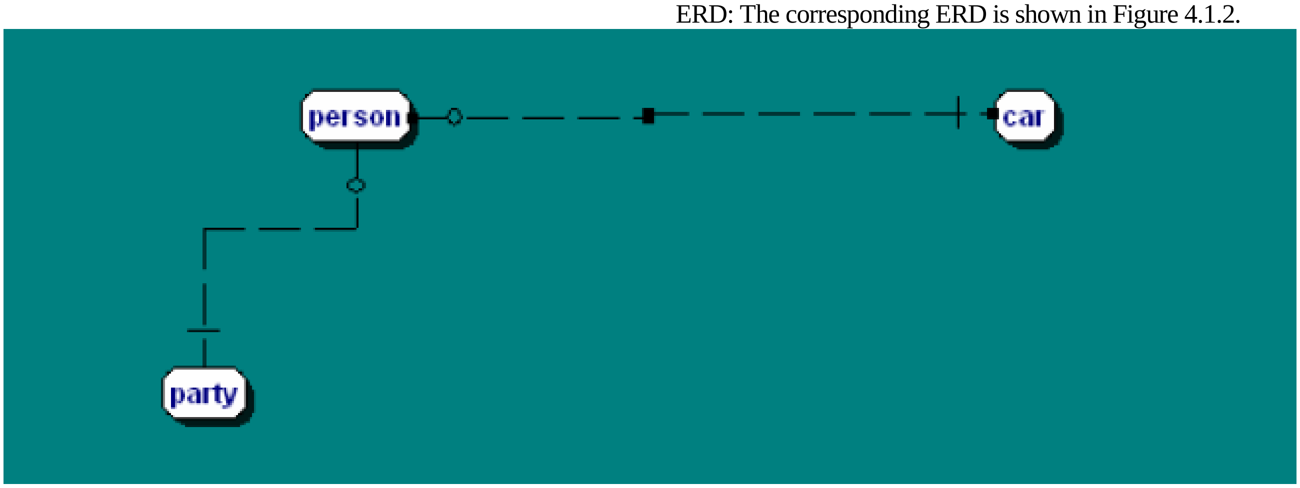

Figure. 4.1.2 - Rule 1

Rule 2

If a noun is followed by has or have and then by noun(s), then: The first noun found is an entity

The second noun(s) found are one or more attributes of the entity.
English Statement: Each department has a unique name and unique number.
Analysis and translation: “Department” is the noun and “name” and “number” are the attributes of department.
ERD: The corresponding ERD is shown in Figure 4.2.

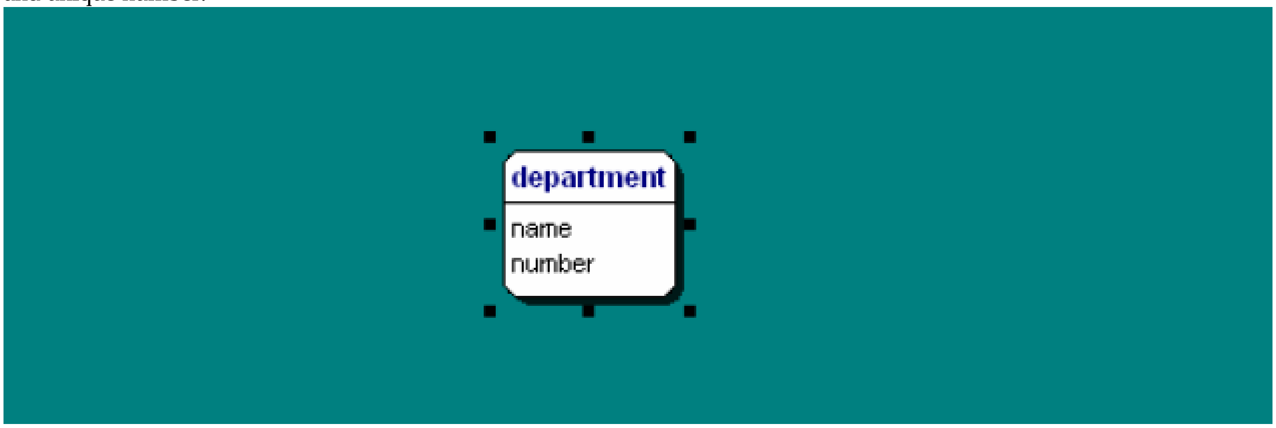

Figure. 4.2 - Rule 2

Rule 3
If a noun is found with an apostrophe ‘s’ followed by other noun then:
a) The first noun is an entity.
b) The ones that follow form attribute of the entity.

English Statement: Each employee’s email and name is stored.
Analysis and translation: “Employee” is the entity and “email” and “name” are its attributes.
ERD: The corresponding ERD is shown in Figure 4.3

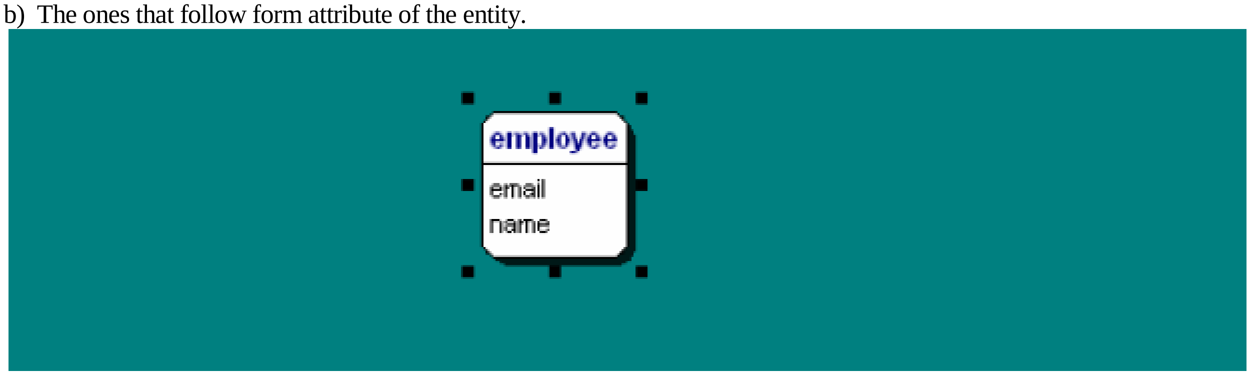

Figure. 4.3 - Rule 3

High-Level Use Cases
The following high level use cases are catered by our tool. These are the ones that are the external actor interacts with.

- Input Problem Statement
- Save Input
- Generate ERD
- Save ERD
- Open ERD
- Export ERD

Analysis and Results
The following calculations were performed for the analysis of the results obtained through the application of above mentioned algorithm.
Text file has been divided into words. Each word contains a tag. This tag is used for the identification of each word i.e. either this word is a noun, proper-noun, verb, adjective etc. Once words have been recognized in a sentence, then algorithm make the sequence of these tagged words. On the basis of this tagged sentence algorithm decides which rule is feasible for this sentence. Now, entities, attributes and relationships have been identified in this sentence. Similarly, this process repeats for each sentence in the text file. As an example, summarized results generated algorithmically in 5 different text files are shown for each entity, attribute and relationship in the Table 1,Table2 and Table 3. The table1 shows the total number of entities manually identified, total number of entities identified by the proposed algorithm, E as O actual entity which were termed as other (attribute or relationship), O as E

actual other (attribute or relationship) termed as a particular entity. The table 2 shows the total number of attributes manually identified, total number of attributes identified by the proposed algorithm, A as O actual entity which were termed as other (entity or relationship), O as A actual other (entity or relationship) termed as a particular attribute. The table 3 shows the total number of relationships manually identified, total number of relationships identified by the proposed algorithm, R as O actual entity which were termed as other (attribute or entity), O as R actual other (attribute or entity) termed as a particular entity. Formulas for calculating Recall and Precision values for entities, attributes and relationships are as follows.

$$\text{Entity Recall} = \frac{\text{Entities Identified Correctly}}{\text{Entities Identified Correctly} + E_{asO}} \tag{1}$$

$$\text{Entity Precision} = \frac{\text{Entities Identified Correctly}}{\text{Entities Identified Correctly} + O_{asE}} \tag{2}$$

Precision defines the proportion of the classified words (Entities/Attributes/Relationships) which are actually correct whereas recall depicts the sensitivity, or the proportion of the correct results obtained.

Table 1 Entity Recall & Precision

File	E(manual)	E(algo)	E as O	O as E	E Recall	E Precicion
1	7	6	1	0	75	85.71%
2	10	8	1	2	72.72%	66.66%
3	14	11	3	0	64.70%	78.57%
4	25	23	1	1	88.46%	88.46%
5	17	15	2	0	78.94%	88.23%

$$\text{Attribute Recall} = \frac{\text{Attributes Identified Correctly}}{\text{Attributes Identified Correctly} + A_{asO}}$$

$$\text{Attribute Precision} = \frac{\text{Attributes Identified Correctly}}{\text{Attributes Identified Correctly} + O_{asA}}$$

Table 2 Attribute Recall & Precision

File	A(manual)	A(algo)	A as O	O as A	A Recall	A Precision
1	19	16	2	1	76.19%	80%
2	26	21	4	1	70%	77.77%
3	29	28	1	0	93.33%	96.55%
4	43	39	3	1	84.78%	88.63%
5	33	27	4	2	72.97%	77.14%

$$\text{Relation Recall} = \frac{\text{Relations Identified Correctly}}{\text{Relations Identified Correctly} + R_{asO}}$$

$$\text{Relation Precision} = \frac{\text{Relations Identified Correctly}}{\text{Entities Identified Correctly} + O_{asR}}$$

Table 3 Relation Recall & Precision

File	R(manual)	R(algo)	R as O	O as R	R Recall	R Precision
1	5	3	2	0	42.85%	60%
2	11	8	1	2	66.66%	61.53%
3	13	9	1	0	64.28%	69.23%
4	17	12	2	3	63.15%	60
5	26	20	5	1	64.51%	74.07%

It has been observed form the table 1, table 2 and table 3 that the accuracy level of entities and attributes identification is very good but the identification of relationships among these entities and attributes is below the satisfactory level. Attributes like id, courseid, deptNo and other short names are the major reason to down the accuracy level of attributes identification. These short names also contribute a lot to decrease the level of identification of relationships among the entities. It has been analyzed that if the writer of the text file uses full words rather than the short words than this accuracy level can be increased up to some satisfactory level. The accuracy for each of the Entities, Attributes and Relationships is 79%, 82% and 63% respectively. Average precision for all of the entities, attributes and relationships is 81.5%, 84% and 67% respectively and average recall for all of the entities, attributes and relations is 76%, 79.4% and 60.2%. Accuracy for the whole system is about 75%.

Future Directions:

The point where we export our text file to the external tool namely DeZign leaves a lot of room for future work in this field. The nature of any such future work can be broadly categorized as follows:

- These rules or guidelines presented are not the extendable. New rules can be added and the presented ones can be modified.
- I have proposed certain constraints and asked the user to give the description to the system that fulfills the constraints. Work can be done to pre process the user description before input to the system, such that the textual input is automatically set according to the constraints.
- The use of semantics rather than structural analysis to help infer many such things that have not been catered e.g. cardinalities, weak attributes, composite attributes etc.
- Implementation of integrity constraints.
- Automated generation of tables that are used by Relational Database Management Systems (RDMS) to implement and maintain databases

Abbreviations

Following is a list of the abbreviations that have been used throughout the documentation.

E = Entity
A-> E = Attribute of Entity
A-> R = Attribute of Relationship
R = Relationship
C = Cardinality
RR = Recursive relationship
ERD = Entity Relationship Diagram
EERD = Extended Entity Relationship Diagram

References

1. Elmasri & Navanthe. Fundamentals of Database Systems, Fourth Edition.
2. http://nl.ijs.si/telri/wg5rep1/node6.html
3. http://www.cs.jhu.edu/~brill/
4. Brill. E. (1994) Some Advances in Transformation Based Part of Speech Tagging, Proceedings of the twelfth national conference on Artificial intelligence (vol. 1) Pages: 722 - 727 ISBN:0-262-61102-3
5. http://www-2.cs.cmu.edu/~benhdj/c_n_s.html
6. Chen, P (2002). Entity-Relationship Modelling: Historical Events, Future Trends, and Lessons Learned. Software Pioneers: Contributions to Software Engineering, 2002.
7. Chen, Thalheim & Wong. Future Directions of Conceptual Modeling. Springer Berlin / Heidelberg, Volume 1565/1999, Pages 287-301 ISBN978-3-540-65926-6
8. Chen, P. P. (1983), `English sentence structure and entity-relationship diagrams', Information Science 29, 127(149).
9. P.P Chen. (1986) The entity-relationship model-A basis for enterprise view of data. Distributed systems, Vol. II: distributed data base systems Pages: 347 – 354 ISBN:0-89006-213-7
10. P.P Chen. (1981) A preliminary framework for Entity-Relationship Models. Proceedings of the Second International Conference on the Entity-Relationship Approach to Information Modeling and Analysis Pages: 19 - 28 ISBN:0-444-86747-3
11. www.datanamic.com/dezign/
12. Albrecht, M., Buchholz, E., D□usterh□oft, A. & Thalheim, B. (1995), An informal and efficient approach for obtaining semantic constraints using sample data and natural language processing., in `Semantics in Databases', pp. 1{28
13. Omar, N., Hanna, P. & Mc Kevitt, P. (2004),

Heuristics-based entity-relationship modeling through natural language processing, in `Fifteenth Irish Conference on Arti_cial Intelligence and Cognitive Science (AICS-04)', pp. 302{313.
14. Syeri Hartmenn, Sebastian Lin.2007, English Sentence Structures and EER Modeling, ACM International Conference Proceeding Series; Vol. 247 Proceedings of the fourth Asia-Pacific conference on Conceptual modeling - Volume 67 Ballarat, Australia Pages: 27 – 35.

12/14/2010