

# EERCASE: Uma Ferramenta para Apoiar o Estudo do Projeto Conceitual de Banco de Dados

Edson Alves, Natália M. Franco, Antônio Nascimento, Robson N. Fidalgo

Centro de Informática – Universidade Federal de Pernambuco (PE)  
CEP 50740-560 – Recife – PE – Brasil

{eas4, nmf, ajnf, rdnf}@cin.ufpe.br

**Resumo.** *Dada a relevância do projeto conceitual de Banco de Dados e da ampla utilização do modelo Enhanced Entity–Relationship (EER), é importante que o aluno construa uma sólida competência sobre os conceitos, técnicas e ferramentas deste domínio. Contudo, nota-se uma carência de ferramentas CASE direcionados ao estudo do modelo EER. Com esta motivação, este artigo apresenta a ferramenta EERCASE como um instrumento pedagógico para auxiliar o ensino do modelo EER. Como diferenciais, a EERCASE apresenta a cobertura de 100% dos construtores do EER e o fornecimento de feedbacks, facilitando a resolução de problemas, reflexão dos resultados e consolidação do conhecimento adquirido.*

## 1. Introdução

Os Curriculums de Computação da ACM/IEEE/AIS [ACM/IEEE 2008][ACM/AIS 2010] recomendam que todo aluno da disciplina básica de Banco de Dados (BD) tenha contato com as três fases do Projeto de um BD, a saber: Conceitual, Lógico e Físico, nesta ordem. Dado que a fase de Projeto Conceitual é a primeira das três fases e tem o maior nível de abstração, é importante que o aluno construa uma sólida competência sobre os conceitos, técnicas e ferramentas desta fase, pois erros nesta são propagados para as demais, o que torna mais complexa e mais custosa a correção destes. Neste contexto, o modelo *Enhanced Entity–Relationship* (EER) [Chen 1976][Elmasri and Navathe 2010] é uma linguagem bem aceita pela comunidade de BD, pois: 1) muitos livros dedicam um ou mais capítulos sobre o assunto; 2) o ensino do modelo EER é recomendado pelos *Curriculums* de Computação da ACM/IEEE/AIS [ACM/IEEE 2008][ACM/AIS 2010] e 3) em se tratando de Projeto Conceitual de BD, o modelo EER é mais expressivo do que a alternativa de usar o Diagrama de Classe da UML [Bavota et al. 2011].

Apesar da notação gráfica do modelo EER ser simples, a atividade de projeto conceitual de um BD não é uma tarefa trivial. Por exemplo, erros sintáticos (i.e., relacionados ao uso incorreto da notação) são recorrentes entre alunos e projetistas iniciantes. Para evitar este tipo de erro, pode-se usar uma ferramenta do tipo *Computer Aided Software Engineering* (CASE), pois estas evitam a construção de esquemas com erros sintáticos. No entanto, como pode ser visto na Seção 4, ainda não existe uma ferramenta CASE que forneça suporte a todos os conceitos da notação EER segundo a visão de Elmasri e Navathe [Elmasri and Navathe 2010] (a notação estendida mais próxima à notação original de Chen [Chen 1976]). Com o objetivo de propor uma solução para este problema, este trabalho apresenta a ferramenta EERCASE (*Enhanced*

*Entity Relationship CASE*) como um instrumento pedagógico para auxiliar professores e alunos em disciplinas de BD da graduação e pós-graduação. Neste contexto, vale ressaltar que a EERCASE: 1) é a primeira ferramenta para fins acadêmicos que cobre 100% os conceitos do modelo EER segundo Chen/Elmasri e Navathe (Ver Quadro 1); e 2) favorece alunos e professores, pois, por se tratar de uma ferramenta CASE, dá condições para melhor entender a resolução de problemas, refletir sobre os resultados obtidos e consolidar o conhecimento adquirido.

O restante deste artigo está organizado da seguinte forma: 1) a Seção 2 aborda o processo de desenvolvimento da ferramenta e foi dividida em metamodelo e tecnologias utilizadas; 2) a Seção 3 apresenta a ferramenta EERCASE com sua interface e modo de utilização; 3) a Seção 4 faz uma avaliação comparativa das soluções relacionadas; e 4) a Seção 5 apresenta o potencial pedagógico da ferramenta e as considerações finais deste trabalho.

## 2. Desenvolvimento da Ferramenta EERCASE

A EERCASE foi desenvolvida seguindo o paradigma *Model-Driven Development* (MDD) [Brambilla et al. 2012], o qual define que uma linguagem de modelagem deve ser especificada a partir de um metamodelo (i.e., a gramática da linguagem de modelagem). As seções a seguir apresentam, respectivamente, uma visão geral do metamodelo da EERCASE e o conjunto de tecnologias usadas para implementar a ferramenta.

### 2.1 Metamodelo

Um metamodelo descreve, sem ambiguidade, a sintaxe abstrata (i.e., gramática) de uma linguagem de modelagem. Isto é, seus construtores e os relacionamentos válidos entre eles. Ou seja, um metamodelo define as regras que especificam o que é um modelo válido [Kelly and Tolvanen 2008]. Na Figura 1, mostra-se o metamodelo *Enhanced Entity Relationship Meta Model* (EERMM) [Fidalgo et al. 2012][Fidalgo et al. 2013] usado no desenvolvimento da ferramenta EERCASE. Vale salientar que, diferente dos dois trabalhos anteriores, a contribuição original deste trabalho é a apresentação da ferramenta EERCASE e não do metamodelo EERMM. Além disso, também é importante destacar que, desde a proposta do modelo ER [Chen 1976], o EERMM e a EERCASE formam a primeira solução computacional que cobre todos os conceitos do modelo EER segundo a notação de Chen/Elmasri e Navathe (ver Quadro 1).

Na Figura 1, um Esquema (*Schema*) é composto por meta-classes do tipo Nó (*Node*) e do tipo Ligação (*Link*). A meta-classe Nó é especializada nas meta-classes: Herança (*Inheritance*), Categoria (*Category*) e Elemento (*Element*), a qual é especializada em Atributo (*Attribute*), Relacionamento (*Relationship*) e Entidade (*Entity*), sendo que esta última é especializada em Entidade Associativa (*AssociativeEntity*). Sobre as meta-classes de Ligação, pode-se observar que estas têm um Nó como origem (*source*) e outro Nó como destino (*target*). Além disso, também pode-se observar que a metaclasses Ligação é especializada nas seguintes meta-classes: Ligação de Atributo (*AttributeLink*), Ligação de Relacionamento (*RelationshipLink*), Ligação de Herança Direta (*DirectInheritanceLink*), Ligação de Generalização, a qual é especializada em Ligação de Generalização de Herança (*InheritanceGL*) e Ligação de

Generalização de Categoria (*CategoryGL*) e, por fim, Ligação de Especialização, a qual é especializada em Ligação de Especialização de Herança (*InheritanceSL*) e Ligação de Especialização de Categoria (*CategorySL*). Uma explicação mais detalhada do EERMM está fora do escopo deste trabalho. Contudo, a especificação completa do EERMM pode ser encontrada em [Fidalgo et al. 2012], [Fidalgo et al. 2013] ou <http://cin.ufpe.br/~eercase/documentation/EERCASE/eercase-package-index.html>.

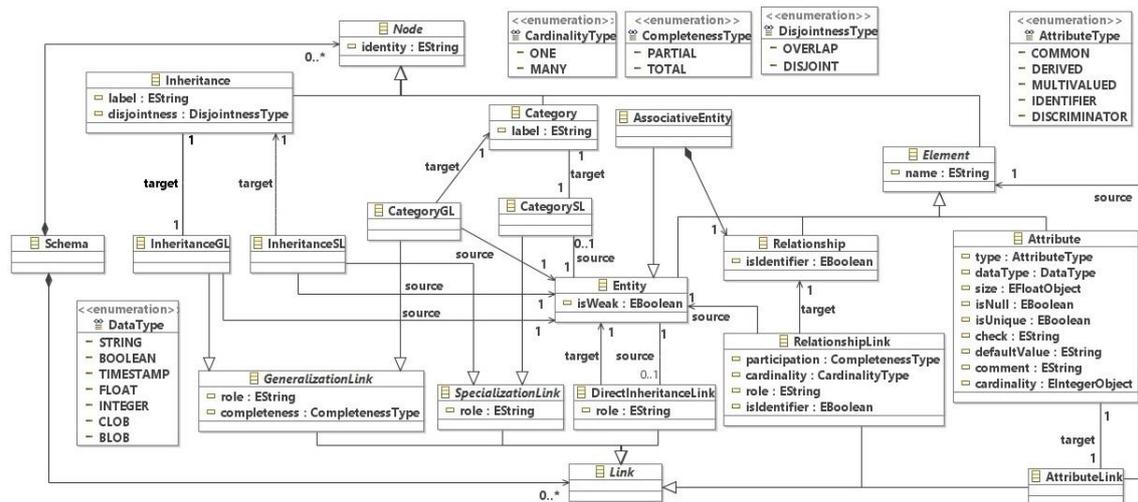


Figura 1 - Metamodelo EERMM

## 2.2 Tecnologias

A EERCASE é implementada usando as seguintes tecnologias: *Graphical Modeling Project* (GMP)[Foundation 2014] (arcabouço de tecnologias básicas para construir ferramentas CASE em *Java/Eclipse*) e *Epsilon Framework* (arcabouço de tecnologias que simplifica a construção de ferramentas CASE em GMP) [Kolovos et al. 2014], as quais estão em conformidade com o padrão *Essential Meta Object Facility* (EMOF) [OMG 2014]. Além disso, utiliza *XML Metadata Interchange* (XMI) para armazenamento, manipulação, recuperação e intercâmbio de metadados. A EERCASE funciona como uma aplicação *standalone* utilizando a arquitetura *Rich Client Platform* (RCP) [McAffer and Lemieux 2005] da plataforma *Eclipse*, o que permite que esta seja independente de plataforma e de fácil distribuição.

O processo de desenvolvimento da EERCASE tem início com a codificação do metamodelo EERMM na linguagem *Emfatic* do *Epsilon Framework*. Vale ressaltar que o código *Emfatic* produzido, além da parte correspondente ao EERMM (i.e., sintaxe abstrata da EERCASE), também contém as especificações gráficas da notação (i.e., sintaxe concreta da EERCASE). Na sequência, a ferramenta *EuGENia* do *Epsilon Framework* é usada para abstrair os detalhes de configuração e execução do GMP. Mais detalhes e tutoriais sobre as tecnologias GPM e *Epsilon Framework* podem ser encontrados em: <http://www.eclipse.org/modeling/gmp/> ou <http://www.eclipse.org/epsilon/>, respectivamente.

### 3. Apresentação da EERCASE

Na Figura 2 é apresentada a interface gráfica da ferramenta EERCASE. Nesta figura é possível notar que: 1) na área “A” tem-se os construtores do modelo EER; 2) na área “B” tem-se a região de desenho do diagrama e 3) na área “C” as propriedades de cada construtor EER podem ser configuradas. A tarefa de modelagem começa com um clique no construtor desejado (área “A”) e depois no local que este deve aparecer na área de desenho (área “B”). Em seguida, o projetista pode editar as propriedades do construtor (área “C”) e adicionar novos construtores. Além desses recursos básicos, a EERCASE tem outros, por exemplo: criação de notas de texto, definição de cores para os construtores, visualização de vários esquemas em abas diferentes, controle de Zoom e desfazer ou refazer uma ação. A ferramenta é distribuída gratuitamente sob a licença *Creative Commons* com atribuição sem derivações (CC BY-ND 2.0 BR). Mais informações sobre a EERCASE (incluindo seu link de download e vídeos demonstrativos) podem ser encontradas em <http://www.cin.ufpe.br/~eercase>.

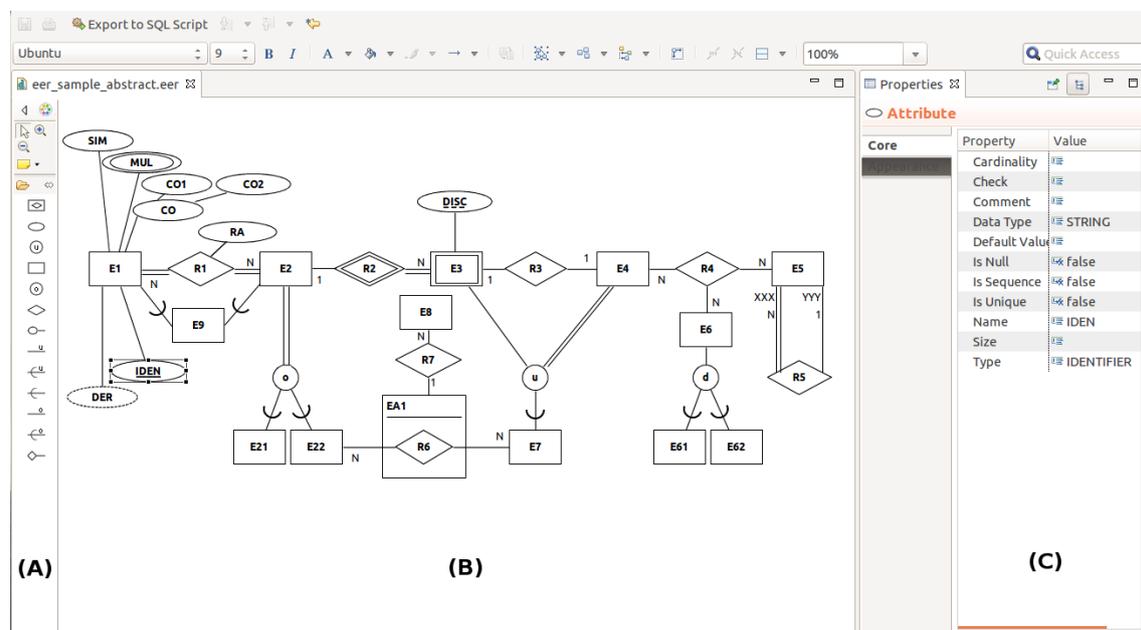


Figura 2- Interface gráfica da EERCASE

Com o propósito de modelar um esquema simples e que cubra todos os construtores EER, na Figura 2 mostra-se um esquema simbólico e independente de domínio. Como pode ser constatado nesta figura, a EERCASE dá suporte para modelar todos os construtores EER segundo Chen/Elmasri e Navathe, a saber: 1) Entidade regular (todas, exceto EA1); 2) Entidade associativa (EA1); 3) Entidade/Relacionamento Fraco (E3); 4) Atributo Simples (todos, exceto CO); 5) Atributo composto (CO); 6) Atributo Multivalorado (MUL); 7) Atributo Derivado (DER); 8) Atributo Identificador (IDEN); 9) Atributo Discriminador (DISC); 10) Relacionamento com atributo (RA); 11) Auto-relacionamento (R5); 12) Relacionamento N-ário (R4); 13) Participação de Relacionamento (linha dupla ou simples); 14) Cardinalidade de Relacionamento ("1" ou "N"); 15) Papeis em Relacionamento (XXX e YYY); 16) Herança Simples (todas, exceto E9); 17) Herança Disjunta ou Sobreposta (um círculo com a letra "d" ou "o", respectivamente); 18) Herança Total ou Parcial (um

círculo com a letra “d” ou “o” ligado a uma linha dupla ou uma linha simples, respectivamente); 19) Herança Múltipla (E9); e 20) Categoria total ou parcial (um círculo com a letra "u" ligado a uma linha dupla ou uma linha simples, respectivamente).

Pelo fato da EERCASE ser baseada no EERMM, esta consegue impedir os seguintes erros sintáticos: 1) uma Ligação de Relacionamento entre Entidades ou entre Relacionamentos, pois uma Ligação de Relacionamento deve ter como origem uma Entidade e como destino um Relacionamento; 2) uma Ligação de Atributo partindo de uma Herança ou Categoria e uma Ligação de Atributo ligando, ao mesmo tempo, um Atributo a mais de uma Entidade, Relacionamento ou outro Atributo, pois uma Ligação de Atributo só pode ter uma Entidade, um Relacionamento ou outro Atributo (Atributo Composto) como origem e um Atributo como alvo; 3) uma Herança com mais de uma Ligação de Generalização, pois uma Ligação de Generalização de uma Herança tem como origem uma única Entidade e como destino uma única Herança; e 4) uma Categoria com mais de uma Ligação de Especialização, pois uma Ligação de Especialização de uma Categoria tem como origem uma única Entidade e como destino uma única Categoria. A Figura 3 ilustra estes erros (marcados com um “x”), bem como a forma de corrigi-los. Vale ressaltar que estes erros sintáticos são básicos e que um estudo sistemático para detectar e impedir erros mais complexos está em andamento.

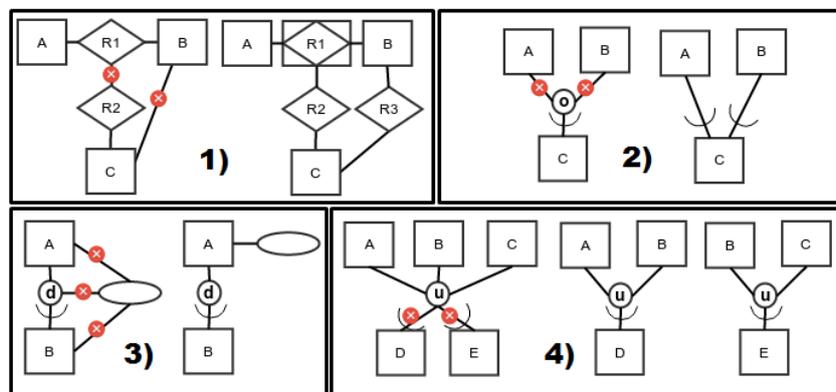


Figura 3: Erros Sintáticos evitados por EERCASE

#### 4. Avaliação Comparativa

Apesar de existirem muitas ferramentas para modelagem de BD, poucas são voltadas para a modelagem conceitual. Isto é, a maioria das ferramentas para projeto de BD tem uma notação gráfica voltada para o projeto lógico do BD, ou seja, não permitem modelar importantes construtores do modelo conceitual, por exemplo: relacionamentos n-ários, atributos em relacionamento, atributos composto, atributo multivalorado, entidade associativa e herança. Neste contexto, buscando avaliar ferramentas CASE para este fim, inicialmente foram consideradas as seguintes opções: brModelo, CA ERwin Data Modeler, DB-Main, DeZign for Databases, Dia, TerraER, ER/Studio XE, Open ModelSphere, Oracle Designer, PowerDesigner, RISE Editor, SmartDraw, e Toad Data Modeler. Contudo, como apenas as ferramentas Open ModelSphere, DB-Main e brModelo oferecem suporte para realizar validação sintática e modelar relacionamentos N-ários e relacionamento com atributos, requisitos importantes para uma ferramenta

CASE de modelagem conceitual de BD, as demais foram descartadas da avaliação. No Quadro 1, mostra-se uma comparação destas ferramentas frente a EERCASE. Ressalta-se que para realizar esta avaliação tentou-se modelar, em cada ferramenta, o esquema simbólico apresentado na Figura 2, pois este cobre todos os construtores do modelo EER segundo a notação de Chen/Elmasri e Navathe.

**Quadro 1 – Análise Comparativa de Ferramentas CASE para EER**

Construções EER	Open ModelSphere	DB-Main	brModelo	EERCASE
1. Entidade Regular	SIM	SIM	SIM	SIM
2. Entidade Associativa	NAO	NAO	SIM	SIM
3. Entidade/Relacionamento Fraco	NAO	NAO	SIM	SIM
4. Atributo Simples	SIM	SIM	SIM	SIM
5. Atributo Composto	NAO	NAO	SIM	SIM
6. Atributo Multivalorado	NAO	SIM	SIM	SIM
7. Atributo Derivado	NAO	NAO	NAO	SIM
8. Atributo Identificador	SIM	SIM	SIM	SIM
9. Atributo Discriminador	NAO	NAO	NAO	SIM
10. Relacionamento com Atributos	SIM	SIM	SIM	SIM
11. Auto-Relacionamento	SIM	SIM	SIM	SIM
12. Relacionamento N-ário	SIM	SIM	SIM	SIM
13. Participação de Relacionamento	SIM	SIM	SIM	SIM
14. Cardinalidade de Relacionamento	SIM	SIM	SIM	SIM
15. Papeis em Relacionamento	SIM	SIM	SIM	SIM
16. Herança Simples	NAO	SIM	SIM	SIM
17. Herança Disjunta ou Sobreposta	NAO	NAO	NAO	SIM
18. Herança Total ou Parcial	NAO	NAO	SIM	SIM
19. Herança Múltipla	NAO	SIM	SIM	SIM
20. Categoria Total ou Parcial	NAO	NAO	NAO	SIM
SIM	45%	60%	80%	100%
NAO	55%	40%	20%	0%

Nas Figuras 4-A e 4-B é mostrado o que foi possível modelar usando as ferramentas Open ModelSphere e DB-Main, respectivamente. Como se pode observar, as notações destas ferramentas são significativamente diferentes da notação de Chen/Elmasri e Navathe. Além disso, também se pode constatar que estas ferramentas não dão suporte para modelar todos os construtores do modelo EER (ver Quadro 1). Já na Figura 4-C tem-se os elementos que puderam ser modelados usando a ferramenta brModelo que, diferente das ferramentas anteriores, possui notação semelhante àquela utilizada por Chen/Elmasri e Navathe. Além disso, a brModelo dá suporte a maioria (80% - ver Quadro 1) dos construtores EER. É importante destacar que as ferramentas DB-Main e brModelo apenas permite uma herança por super-entidade, o que é uma restrição grave, pois impede a modelagem de duas ou mais heranças (e.g., uma herança disjunta e outra sobreposta) a partir de uma mesma super-entidade.

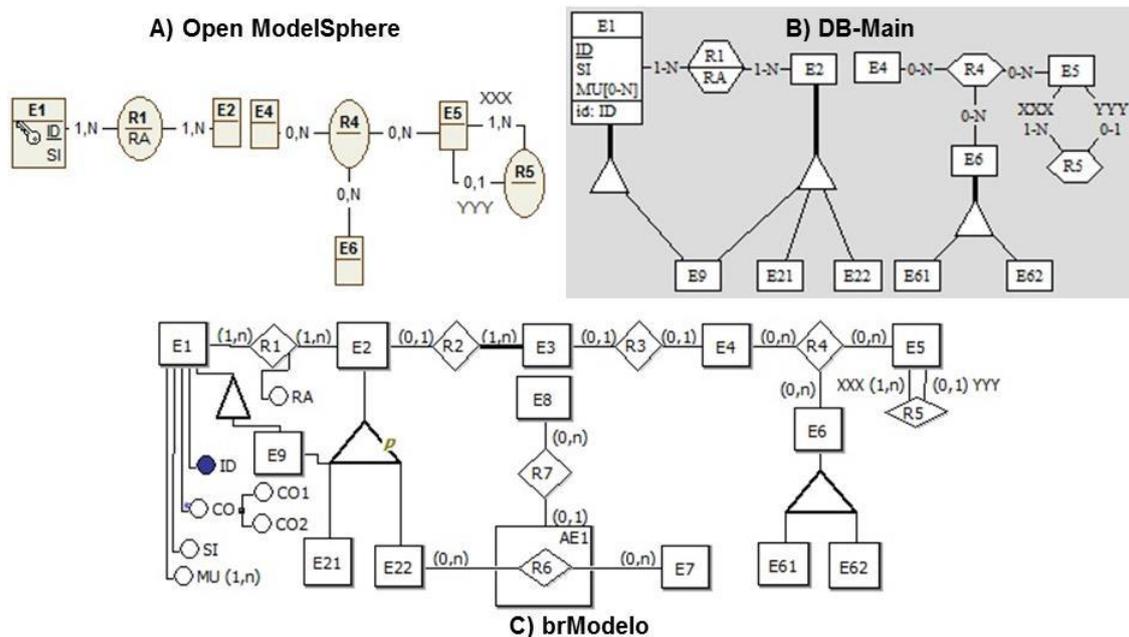


Figura 4 – Avaliação das Ferramentas

## 5. Considerações finais

Dada a relevância e complexidade da fase de projeto conceitual de BD, ferramentas CASE que apoiem o estudo do modelo EER são importantes instrumentos pedagógicos. Neste sentido, este artigo apresentou a ferramenta EERCASE, a qual tem os seguintes pontos fortes: 1) maior aderência ao modelo EER de Chen/Elmasri e Navathe – permite que o aluno pratique todos os conceitos do modelo EER; e 2) *feedbacks* interativos – por ser baseado no metamodelo EERMM, a ferramenta impede a modelagem de diagramas com erros sintáticos, exibindo um sinal de proibido, o que atua como sinalizador para que o aluno perceba que algum conceito ainda não foi bem compreendido. Ademais, o usuário tem conhecimento das propriedades de cada um dos construtores do EER (Área “C” da Figura 2), o que permite uma abordagem construtivista [Piaget 2002] do conhecimento, visto que, neste caso, o conhecimento é adquirido por meio de interações entre o indivíduo (aluno) e o meio (ferramenta). Vale salientar que a EERCASE não atua sozinha no processo de aprendizagem, ela funciona como uma ferramenta auxiliar para professores e alunos. Além disso, também é importante destacar que a EERCASE: 1) é reconhecida como um caso de sucesso do *Epsilon Framework* (<http://eclipse.org/epsilon/users/>); 2) é usada como ferramenta padrão para o ensino das disciplinas básicas de BD (graduação e pós-graduação) da UFPE; e 3) que esta tem boa aceitação por parte dos usuários (professores e alunos), pois é a primeira ferramenta que faz validação sintática e abrange todos os conceitos do modelo EER segundo Chen/Elmasri e Navathe. Por fim, espera-se que a EERCASE seja adotada por outras instituições de ensino. Como trabalho futuro, está em andamento a implementação de um módulo para validações mais avançadas e a geração automática de *scripts* SQL/DDL.

## Referencias

- ACM/IEEE (2008). Computer Science Curriculum 2008: an interim revision of CS 2001. <http://www.acm.org/education/curricula/ComputerScience2008.pd>.
- ACM/AIS (2010). Curriculum guidelines for undergraduate degree programs in information technology. [http://www.acm.org/education/curricula/IS\\_2010\\_ACM\\_final.pdf](http://www.acm.org/education/curricula/IS_2010_ACM_final.pdf).
- Bavota, G., Gravino, C., Oliveto, R., De Lucia, A., Tortora, G., Genero, M., and Cruz-Lemus, J. (2011). Identifying the weaknesses of uml class diagrams during data model comprehension. In Whittle, J., Clark, T., and Kuhne, T., editors, *Model Driven Engineering Languages and Systems*, volume 6981 of *Lecture Notes in Computer Science*, pages 168–182. Springer Berlin Heidelberg.
- Brambilla, M., Cabot, J., and Wimmer, M. (2012). *Model-Driven Software Engineering in Practice*. *Synthesis Lectures on Software Engineering*. Morgan Claypool Publishers.
- Chen, P. P.-S. (1976). The Entity-Relationship Model - toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36.
- Elmasri, R. and Navathe, S. (2010). *Fundamentals of Database Systems*. Prentice Hall International, 6 edition.
- Fidalgo, R. D. N., Alves, E., España, S., Castro, J., and Pastor, O. (2013). Metamodeling the enhanced entity-relationship model. *JIDM*, 4(3):406–420.
- Fidalgo, R. D. N., De Souza, E., España, S., De Castro, J., and Pastor, O. (2012). Eermm: A metamodel for the enhanced entity-relationship model. In Atzeni, P., Cheung, D., and Ram, S., editors, *Conceptual Modeling*, volume 7532 of *Lecture Notes in Computer Science*, pages 515–524. Springer Berlin Heidelberg.
- Foundation, E. (2014). *Graphical Modeling Project (GMP)*. <http://www.eclipse.org/modeling/gmp/>.
- Kelly, S. and Tolvanen, J.-P. (2008). *Domain-Specific Modeling: enabling full code generation*. Wiley-IEEE Computer Society Pr.
- Kolovos, D., Rose, L., García-Domínguez, A., and Paige, R. (2014). *The Epsilon Book*. <http://www.eclipse.org/epsilon/doc/book/>.
- McAffer, J. and Lemieux, J.-M. (2005). *Eclipse Rich Client Platform: Designing, Coding, and Packaging Java(TM) Applications*. Addison-Wesley Professional.
- OMG (2014). *Essential Meta-Object Facility (EMOF)*. <http://www.omg.org/spec/mof/2.4.1/pdf>.
- Piaget, J. (2002). *Epistemologia genética*. Psicologia e pedagogia. Martins Fontes.