ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 8, Issue 6, December 2018

Animating the Mapping Process for Converting an ERD to Relations

Lu Zhang
School of Engineering and Computing, National University, San Diego, CA 92123, USA

***Abstract*— In teaching the subject of conceptual modeling, it is found that many students have difficulty fully comprehending the mapping process for converting an Entity Relationship Diagram (ERD) into its corresponding set of relations. This is because the traditional way of teaching the mapping process involves "too much math," as characterized by struggling students. In a previous work, the author, along with other colleagues, proposed a different teaching method that utilizes task maps, visual clues, and animations to "un-math" the complexity perceived by students. The details of task maps and visual clues were described and illustrated in the previous work. However, the subject of the actual implementation of 3D animations that enhance students' learning by turning abstractions into an animated environment was laid out as future research work. This work-in-progress paper is the continuation of the overall effort for the new teaching approach and reports the status of the development of the animations and the experience in this endeavor. It also discusses related future work.**

***Index Terms*— Relational Databases, Relations, Tables, Entity Relationship Diagrams, Animations.**

I. INTRODUCTION

Based on the author's experience in teaching database classes, many students have difficulty mastering the mapping process for converting an Entity Relationship Diagram into its corresponding relational schema. Specifically, students seem to have trouble comprehending the overall picture and context of the mapping process and rules. They also have difficulty discerning the arrangement of primary keys based on the cardinalities of relationships. This includes deciding when it is necessary to create new relations during the mapping process. This perhaps is surprising to many since the mechanism for the mapping process is not overwhelmingly complicated. Further exploration reveals that the traditional way of describing and teaching the mapping process, as perceived by students, involves "too much math." This not only forges a mental barrier hindering students from learning the mapping concepts properly and effectively but also discourages students from even trying. Therefore, the first author, along with other colleagues, proposed a different approach to teach the mapping process that "un-maths" the math bias associated with the traditional approach [12]. The proposed approach is visual-based, as most people are visual learners, and is summarized in this section to provide the necessary relevant context.

Manuscript received: 20 November 2018
Manuscript received in revised form: 17 December 2018
Manuscript accepted: 03 January 2019
Manuscript Available online: 10 January 2019

The new approach utilizes task maps, visual clues, and animations. Task maps, shown in Figure 1, provide an overall roadmap and context of the mapping process. Further, visual clues are embedded into both task maps (Figure 2) and mapping rules (Figure 3) to eliminate seemingly complex mathematical notations.

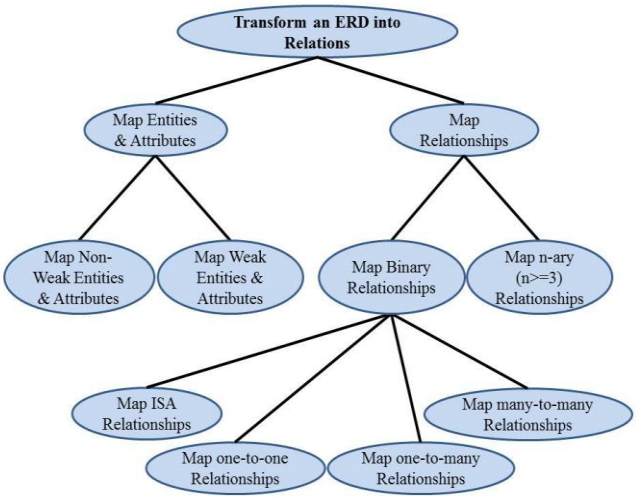


Fig 1. Task map outlining the mapping process

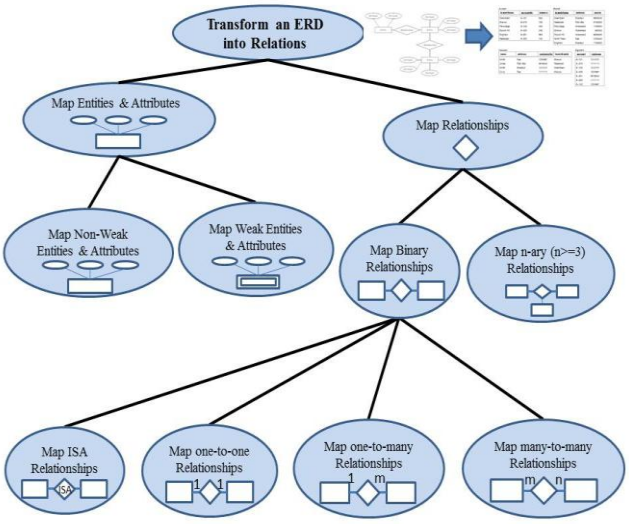


Fig 2. Task map with Visual Clues

The key component of this approach is to use animations to enhance students' learning by turning abstractive mapping rules into an animated environment. Animation has always been an integral part of Computer Science education. For example, Baecker [2] produced a 30-minute video from 1978 to 1981, entitled Sorting out Sorting that used animation with voice-over explanatory narratives to teach sorting algorithms. In addition to helping Computer Science students [13], animations have also been used to help science

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 8, Issue 6, December 2018

students understand abstract concepts and invisible processes (e.g., [10], and [11]).

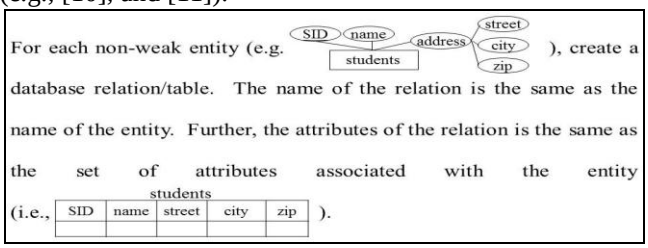


Fig 3. Mapping Rule #1 with Visual Clues

Well-designed educational animations can bring conceptual processes and abstract concepts to life "by graphically representing their various states and animating the transitions between those states" [7]. Animations can thus have a positive impact on understanding abstract concepts and processes by revealing interesting visual clues for the learning process.

It has been suggested in [4] that "computer animation is good for depicting dynamic content because animations can show changes over time (temporal changes) which is especially useful for teaching processes and procedures." Further, "well-designed animations may help students learn faster and easier. They are also excellent aid to teachers when it comes to explaining difficult subjects. The difficulty of subjects may arise due to the involvement of mathematics or imagination" [4]. Also, researchers at the University of Kentucky Center for Visualization and Virtual Environments have studied the effects of animations in supporting learning processes and find that visual representations are superior to text-based representation and reduce students' cognitive load, but only in certain tasks where relationships are discussed" [1].

The mapping process renders itself as a perfect candidate for animation. The task of creating tables based on an ERD requires the ability to relate entities with their corresponding tables, that is, understanding the relationships between the two. This can also be dynamically depicted with animations. Animations can also demonstrate temporal changes of the resulting tables by highlighting the movement of primary keys of the involved entities based on the cardinalities of the associated relationships and illustrating what the resulting relations should look like.

It is necessary to point out that the assertion that visualization makes learning easier has been challenged "because the evidence is sometimes equivocal and frequently unclear on whether these new heights achieved through powerful computers actually enhance learning" [8]. Similar observations have also been made (i.e., [4]) that "research evidence about the educational effectiveness of animations is mixed." Some studies have found animations to have a positive impact on learning while others have found otherwise or even negative effects (i.e.[4]). However, it has also been observed [7] that the field of research on visualization has naturally been dominated by cognitive psychologists using tightly designed "control" experiments, based on the manipulation of specific "variables" with undergraduate psychology students as the subjects. Consequently, the outcome "is not transferable to an understanding of the messy world of the everyday learning of established knowledge by science students." There is evidence, as indicated earlier in this paper, that animations have a positive impact on STEM education. It should be obvious that assessment is an important and integral part of this effort, which will be elaborated in the conclusion section of this paper.

II. METHODS

The Alice programming environment, designed by Carnegie Mellon University, is used to create animations to demonstrate the mapping process for converting an entity relationship diagram to a set of database tables. Alice uses the Java platform and is object-oriented. According to its official website, "Alice is an innovative block-based programming environment that makes it easy to create animations, build interactive narratives, or program simple games in 3D."

Alice provides a 3D programming environment for creating animations. It uses drag-and-drop to populate a virtual world with 3D objects (e.g., people, animals, etc.) to create animations that tell a story. The interactive programming environment is designed to be easy to use, even for people who are unfamiliar with programming. Alice also has all the required features for generating effective animations for the mapping rules. For example, it has been observed (i.e., [4]) that the pace of animation may exceed learners' ability to process information. Therefore, a good educational animation should allow learners to control how it plays [4] with the ability to speed up and slow down and to zoom in and out so that attention to details can be drawn to "reveal new relationships or processes that otherwise might have been missed." [8]

A group of undergraduate students in the Information Systems program at National University was enlisted to implement these animations. Even though these students do not have prior experience in building animations or games, they have been selected as many of them are enthusiastic computer game players. In addition, a high-school student was engaged to carry out a separate implementation. One of the purposes for doing so is to encourage students in secondary school to pursue STEM education. Another intention is to create multiple animation series that can be adopted in classrooms to compare their effectiveness.

III. RESULTS

There are a number of schemes presented in literature (e.g., [3], [5], and [6]) for converting an ERD into its corresponding set of relations. The mapping process has been summarized into eight rules [12]. The rules are

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 8, Issue 6, December 2018

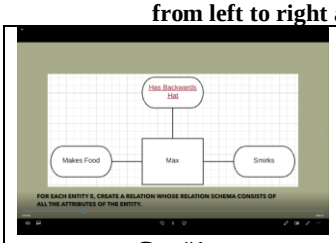
organized as follows: (1) one rule for mapping regular (non-weak) entities, (2) one rule for mapping weak entities, (3) four rules for converting binary relationships (one for each type of binary relationships, i.e., ISA, one-to-one, one-to-many, and many-to-many), (4) one rule for mapping n-ary relationships with n >= 3, and (5) one rule for mapping multi-valued attributes. For the creation of animations, this eight-rule process has been simplified into 7 rules. Specifically, the rule for mapping multi-valued attributes has been eliminated since a multi-valued attribute can be modeled with a one-to-many relationship. Therefore, it is assumed that multi-valued attributes are represented using one-to-many relationships on ERDs. The seven rules are summarized in the table below.

Table 1: Mapping Rules

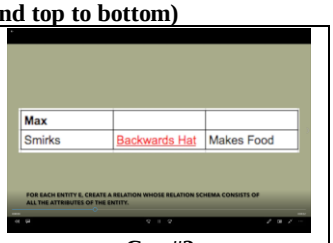
IV. Rule #1	V. For each non-weak entity *E*, create a relation/table *R*. The name of the relation *R* is the same as the name of the entity *E*. Further, the attributes of the relation is the set of attributes associated with the entity plus all the simple component attributes associated with their composite attributes of *E*.
VI. Rule #2	VII. For each entity *A*, which is related to another entity *B* via an "ISA" relationship (i.e., *A* "ISA" *B*), include in the relation corresponding to *A* the primary key of *B*.
VIII. Rule #3	IX. For each binary *one-to-one* relationship *R* between entities *A* and *B* with their corresponding relations *S* and *T*, include in *S*, the primary key of *B*. Further, if the relationship *R* has attributes, include them in *S*. Alternatively, choose *T* in the role of *S*.
X. Rule #4	XI. For each binary *one-to-many* relationship *R* between entities *A (1-side)* and *B (n-side)* with their corresponding relations *S* and *T*, include in *T*, the primary key of *A*. Further, if the relationship *R* has attributes, include them in *T*.
XII. Rule #5	XIII. For each binary *many-to-many* relationship *R* between entities *A* and *B* with their corresponding relations *S* and *T*, create a new relation *Q* with the same name as the relationship *R* and include in *Q*, the primary key of *A* and *B*. Further, if the relationship *R* has attributes, include them in *Q*.
XIV. Rule #6	XV. For each *n*-ary (*n*>=3) relationship *R*, create a new relation *Q* and include in *Q* the primary keys of all the entities involved in *R*. Further, if the relationship *R* has attributes, include them in *Q*.
XVI. Rule #7	XVII. For each weak entity *E*, create a relation *R* whose schema consists of all the attributes of the entity *E* plus the partial key attributes of weak entity *E*'s owner entity or entities. *E*'s owner entities are those entities connected to entity *E* via identifying relationships (double-diamonds).

Up to this point, the undergraduate team has implemented the rules to convert entities and attributes, ISA relationships, 1:1, and 1:m relationships to their corresponding tables (e.g., Rules 1-4). All the animations feature a main character named Max. Max's key characteristic (i.e., the primary key) is that he wears a hat. In the implementation of each rule, the animation starts with the appropriate context by explaining the rule that's being animated, as shown in Cut #1 and Cut #2 in Table 2, followed by the actual animation content. For rule #1, a table-like object (i.e., the white object in Cut #3) has been used to represent the entity that eventually turned into a table/relation (i.e., Cut #9, #10, and #11 in Table 2) after all its attributes have been assembled (i.e., Cut #3, #4, #5, #6, #7, and #8 in Table 2.) The animation finishes with the context to reinforce. Collecting all the attributes in the animation exhibits the fact that the attributes of the relation corresponding to an entity is composed of the attributes belonging to the underlying entity itself.

Table 2: Main Animation Sequence for Rule #1 (sequenced from left to right and top to bottom)



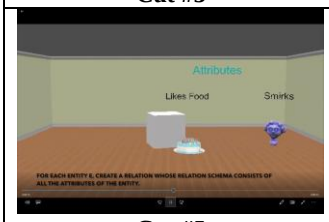
Cut #1



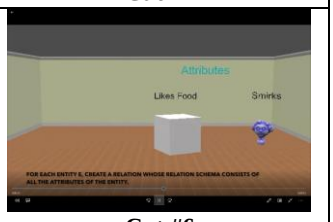
Cut #2

Cut #3

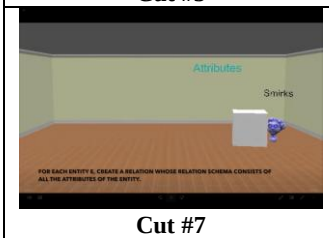
Cut #4



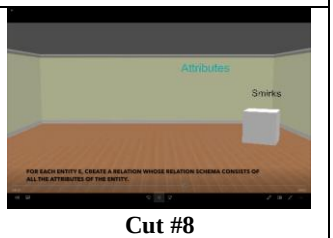
Cut #5



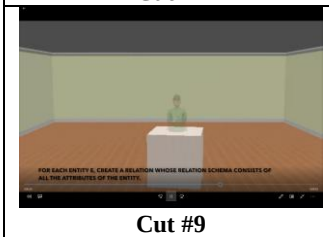
Cut #6

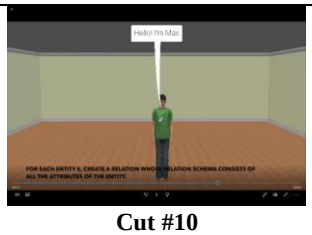


Cut #7

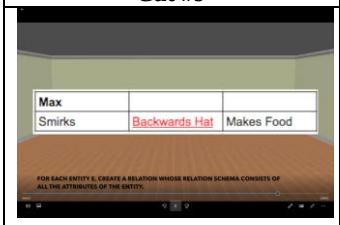


Cut #8

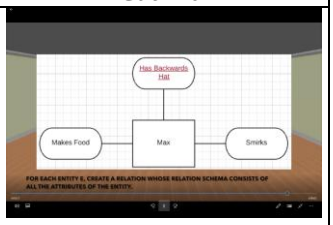

Cut #9



Cut #10



Cut #11

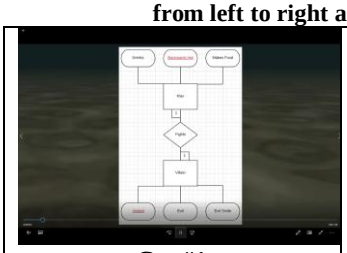


Cut #12

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 8, Issue 6, December 2018

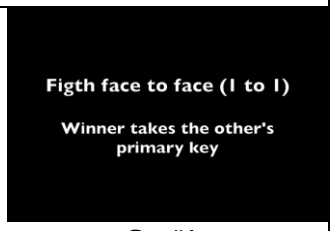
Due to space limitation, the implementation of Rule #3 will be discussed next since it is similar to that of Rules #2 and #4. For Rule #3, in addition to Max, the Villain character is added to the animation to establish a 1:1 relationship between the two. The Villain's key characteristic (i.e., the primary key) is that he is violent. The context is shown in Cuts #1 to #3 in Table 3. For a 1:1 relationship, for example, between entities A and B, either the relation corresponding to A takes the primary attributes of entity B or vice versa. This is visualized in the animation as a fight between Max and the Villain where the winner takes the primary key of the loser (Cuts #4 to #7).

Table 3: Main Animation Sequence for Rule #3 (sequenced from left to right and top to bottom)


Cut #1

Cut #2

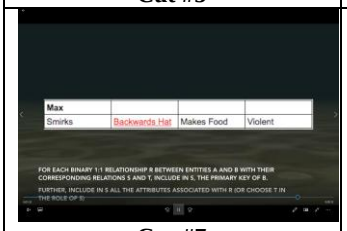
Cut #3



Cut #4


Cut #5

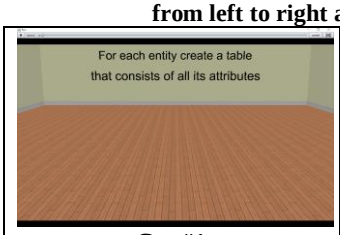

Cut #6



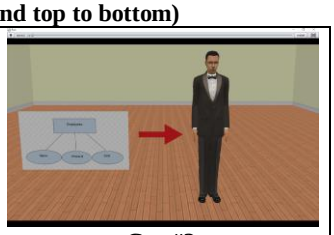
Cut #7

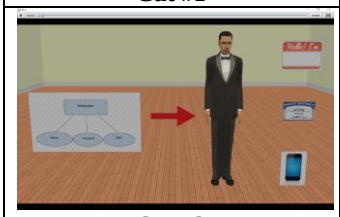
The high-school student, on the other hand, has implemented Rules #1 and #2. Table 4 below shows the main animation segments for Rule #1. In a similar fashion, the animation starts with the context of the rule (i.e., Cuts #1 and #2 in Table 4). Further, a voice-over narrative is utilized to explain the animation of the rule being implemented. The person (i.e., the employee) in Cut #2 is used as a transitional object. The animation sequence (Cuts #3 and #4) of the person "grabbing" the attributes associated with the employee's entity (i.e., name tag, security card, and phone) depicts the transformation of the employee's entity into the employee's table (Cut #5).

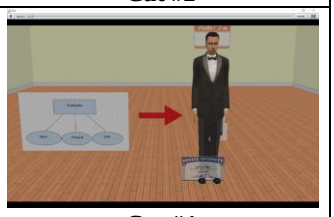
Table 4: Main Animation Sequence for Rule #1 (sequenced from left to right and top to bottom)

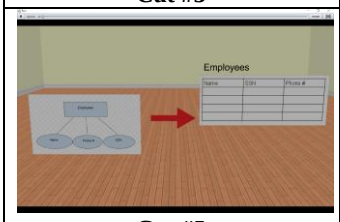


Cut #1


Cut #2


Cut #3


Cut #4



Cut #5

IV. CONCLUSION AND DISCUSSION

The implementation effort of the animations based on the author's previous work for transforming an ERD to its corresponding database tables has been discussed. Alice has the capability to create animations using a relatively simple user interface that provides many built-in props and models. Although these are the only objects that can be easily utilized to create an animation environment, since using an outside image would result in a lot of additional work, Alice, overall, has been found to be adequate. Other programming environment and languages can make much more complex and intricate products, Alice can still be utilized to create desired animations for illustrating the mapping process. Not surprisingly, Alice still bears bugs and inefficiencies. However, workarounds are always available to circumvent them. The experience in this endeavor has been quite positive so far. The simple drag-and-drop environment is very user friendly. Alice, indeed, has some qualities that demonstrate its benefits.

One of the major hurdles in using animations in classrooms is the vast amount of workload required to develop them. It should be noted that our overall research effort has taken a big step forward. As reported, the undergraduate team has completed the animations for Rules 1-4, while the high-school student has delivered Rules #1 and #2. The development effort will continue until all the remaining mapping rules (i.e., Rules 5-7) have been animated.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 8, Issue 6, December 2018

As pointed out earlier, there have been mixed reports on the effectiveness of animations on education in general. Therefore, the effectiveness of this new approach for teaching the mapping process will be assessed. To that end, assessment exercises will be designed, and classroom activities will be carried out so that the effectiveness of using animation as a tool for teaching the mapping process can be evaluated.

The animations produced by the undergraduate team and the high school student are different. The implementation by the undergraduate students employs more animated content than the one from the high-school student. On the other hand, the arrangement produced by the high school student is straightforward and utilizes a voice-over narrative that depicts the rule being animated. It has been recommended (e.g., [8]) that educational animations should be simple as visual overload can be distractive and impair learning. Therefore, it will be interesting to gauge the effectiveness of the two different implementations.

Some studies ascertain that animations can improve students' motivation to learn (e.g., [8], [9], and [11]). Therefore, questionnaires will also be developed to shed some light on this.

REFERENCES

[1] Animations and Learning (n.d.), In "The University of Kentucky Center for Visualization & Virtual Environments," Retrieved from http://vis.uky.edu/research/user-experience/visual-interfaces-learning/animations-and-learning/.

[2] Baecker, R., "Sorting Out Sorting: A Case Study of Software Visualization for Teaching Computer Science," Software Visualization: Programming as a Multimedia Experience. MIT Press, pp. 369-381, 1998.

[3] Coronel, C., Morris, S., & Rob, P., "Database Systems: Design, Implementation, and Management," Cengage Learning, ISBN-13: 9781111969608, 2012.

[4] Education Animations, "Institute of Progressive Education and Learning," Retrieved from http://institute-of-progressive-education-and-learning.org/elearning-i/elearning-educational-entertainment/education-animations/.

[5] Elmasri, R., & Navathe, S., "Fundamentals of Database Systems", Addison-Wesley, ISBN-13: 9780136086209, 2011.

[6] Garcia-Molina, H., Ullman, J., & Widom, J., "Database Systems: The Complete Book", Pearson Education, ISBN-13: 9780131873254, 2014.

[7] Gilbert, J. K., "The role of visual representations in the learning and teaching of science: An introduction," Asia-Pacific Forum on Science Learning and Teaching, Volume 11, Issue 1, 2010.

[8] Phillips, L. M., Norris, S. P., & Macnab, J. S., "Visualization in Mathematics," Reading and Science Education. Springer, ISBN-13: 978-9048188154, 2010.

[9] Rieber, L. P., "Animation, incidental learning and continuing motivation," Journal of Educational Psychology, 83: 318-328, 1991.

[10] Sanger, M. J, Brecheisen, D. M, Hynek, B., "Can Computer Animations Affect College Biology Students' Conceptions about Diffusion and Osmosis?" The American Biology teacher, 63 (2): 104-109, 2001.

[11] Soika, K., Reiska, P, & Mikser, R., "The Importance of Animation as a Visual Method in Learning Chemistry," Proceedings of Fourth International Conference on Concept Mapping, Viña del Mar, Chile, 2010.

[12] Zhang, Lu, Wyne, Mudasser, Farahani, Alireza, Sinha, Bhaskar, Amin, Mohannad, "Visual Learning Tool for Teaching Entity Relationship Mapping Rules," *American* Society for Engineering Education 2015 Pacific Southwest Conference, April 10-11, 2015, San Diego, USA.

[13] Zulhisam, M., & , Riaza, R., "Learning Binary Tree Algorithm Using 3-D Visualization: An Early Results," Procedia - Social and Behavioral Sciences, *90*, 388-395, 2013.

AUTHOR BIOGRAPHY

Lu Zhang Dr. Zhang is a professor at National University in the School of Engineering and Computing at 3678 Aero Court, San Diego, CA 92123. His main research interests include science and engineering education, database technologies, leadership, and strategic management. Dr. Zhang received his Ph.D. in Computer Science from Iowa State University. He is the program director for the B.S. in Information Systems program at National University. He teaches both undergraduate and graduate in computer science, information systems, and data analytics. He has a special interest in behaviors in multicultural settings. He is a member of Delta Mu Delta and an honorary member of Sigma Beta Delta.