ASEE 2014 Zone I Conference, April 3-5, 2014, University of Bridgeport, Bridgeport, CT, USA.

Dynamics of a Cross-disciplinary Corporate-Sponsored Undergraduate Computer Science Project

Thomas Goddard, Konstantin Litovskiy,
Nathan Nichols-Roy, Matthew Reed,
Igor Shvartser, Nicholas Smith, and David Zeppa,
University of California, Santa Cruz
Santa Cruz, CA

with Linda Werner, Ph.D.,
University of California, Santa Cruz, CA, 95064

Julia E. Rice, Ph.D., Hans W. Horn, Ph.D.,
and Amanda C. Engler, Ph.D.,
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95210

Abstract — **This paper focuses on the interactions between an undergraduate student computer science group, an undergraduate chemistry student, a computer science professor, and industry leaders in computational chemistry. We discuss our initial experiences to build a cross-disciplinary project team whose goal is to contribute to an open source software tool used by scientists for polymer discovery.**

I. Introduction

ACCELERATED material discovery is at the core of innovation, economic opportunities, and global competitiveness. The research process is responsible for bringing new materials to market. In 2011, President Obama launched the U.S. Material Genome Initiative (MGI) and challenged researchers, policy makers, and business leaders to reduce the time and resources needed to bring new materials to market—a process that today can take 20 years or more. There is great potential in leveraging modern data mining, big data analytics techniques, and physics based modeling (high performance computing) to significantly shorten the Research & Development cycle in material sciences. Polymers, as an important part of materials science, are the focus of many research fields such as semiconductors (e.g. low-k dielectrics, photolithography, and directed self-assembly), nanomaterials, polymeric drug delivery vehicles, desalination membranes, and recyclable polymers for green chemistry.

Often research is limited by technological constraints (e.g., no automation exists for identification of polymers). Our project's goal is to identify polymers in published research. The purpose of this paper is to describe our initial experiences building a cross-disciplinary project team[1] to address the project goal: identifying polymers in published research. This work is part of a two-quarter long software engineering project; the student team consists of computer science, math, and chemistry majors at the University of California, Santa Cruz (UCSC).

Three researchers at IBM Almaden Research Center in San Jose, CA, mentored the undergraduate team. These researchers are part of a larger group at IBM Research that has considerable expertise in the experimental development and computational modeling of polymers. However, it is tedious to collate data from the literature. Ultimately, chemists would like to initiate a polymer design project by asking questions such as "What polymers make good drug delivery vehicles?", "What are their relevant properties?", and "What are their known problems?" With technology available today, there is no simple method to extract polymer information from patents and journal articles, particularly, extracting polymer structure diagrams from these documents in a meaningful way.

[1] We refer to the entire team, including undergraduate students and corporate research scientists as 'the team.' We refer to the undergraduate student portion of the team as the 'student team.' We refer to the student team without the chemistry student as the 'initial student team.'

The project expands the open source tool OSRA (Optical Structure Recognition Application) [i,ii], currently available for extracting chemical structures from text, to extracting polymer structures from text. The simplest polymers are chain polymers with a single “repeat” unit, specified by $(..)_n$ such as in the image of the polytrimethylenecarbonate illustrated in Figure 1. Today, OSRA can process the ethylene glycol molecule (Figure 2), that is the one without $(..)_n$, but not the polymer.

II. Challenges of Cross-disciplinary Collaboration

There are many challenges in cross-disciplinary collaborations [iii]. Our challenges included lack of understanding of fundamentals in both computer science and chemistry, lack of a common vocabulary, distribution of the team over multiple sites, and team scheduling conflicts. Fortunately, we did not need to face the challenge of dealing with intellectual property (IP) rights, and/or non-disclosure agreements (NDAs), since our corporate sponsors decided before the project’s launch to work with OSRA, an open source software package. We will now describe each of the challenges that we had to face, and the approaches we used to address and overcome them.

A. Lack of understanding of fundamentals of cross-disciplinary sciences

In order for the team to cooperate and perform effectively, all members needed to have at least a fundamental understanding of both the chemistry and computer science relevant to the project. The chemists needed to be convinced that the computer science components, as chosen by the initial student team, appropriately mapped to the application problem (e.g., the data structure accurately represented a general polymer). In addition, the computer science students needed to understand molecular structure diagrams in order to model them in software.

Details of the background needed to start the project were itemized for both subgroups (the initial student team and the IBM researchers). In response, some fundamental questions were raised. For example, how much knowledge did the students and the IBM advisors know about their colleagues’ fields, and how much would the team need to know to advance?

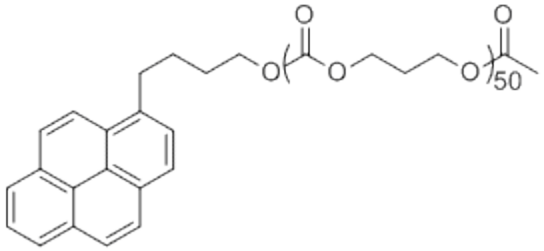


Fig. 1. Polycarbonate-TMC-50

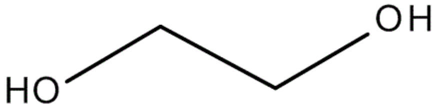


Fig. 2. Ethylene Glycol

Details of the background needed to start the project were itemized for both subgroups (the initial student team and the IBM researchers). In response, some fundamental questions were raised. For example, how much knowledge did the students and the IBM advisors know about their colleagues’ fields, and how much would the team need to know to advance?

As computer science, math, and engineering majors, the members of the initial student team possessed only a high-school level chemistry background and so were expected to acquire a preliminary understanding of organic chemistry before being given an in-depth explanation of the project specifics. We accomplished this by individual study of online sources [iv]. For most of the initial student team, this served to refresh our understanding of the basic properties of molecules and chemical bonds, as well as provide an introduction to the various styles of organic chemistry structure notation. However, when we actually began work on the project, we found that in-person, crash-course sessions, were also necessary in order to understand the material in the context of the project. Fortunately, several sessions were arranged with the IBM research team that covered polymer notation, SMILES strings [v,vi,vii], and SDF (Structure Data format) files [viii]. This was also a helpful time to understand requirements of the project in detail and consider the functionality and limitations of the OSRA software.

In order for the chemists to understand the implementation process, the student team prepared a presentation that outlined the functionality of the software at a high level [ix]. This allowed the student team to explain implementation strategies and receive feedback from the chemists. Specifically, the student team explained the process of converting a bitmap image to a vector image, and how points used to describe curves in the vector could, potentially, be useful for identifying different parts of a chemical diagram. Fig. 3 shows the unique vector paths in different colors.

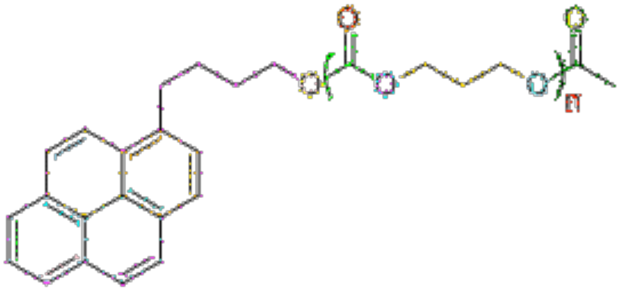


Fig. 3. Vector paths on a diagram

In addition to learning background material, the student team decided that it would be beneficial to maintain contact with an intermediary to assist with answering chemistry related questions that arose during the implementation phase. The course professor sought a student with an organic chemistry background to assist with the initial student team by serving as an intermediary between the student team and the research scientists. Initially, emails were the primary communication between the professor and the intermediary. This was followed by an interview, including a discussion of the team's goals, and how the benefits of cross-disciplinary collaboration could lead to a more effective team. The intermediary was assigned to the project and works directly with the professor, once a week, to discuss approaches on how to become a more resourceful consultant for the student team. Additionally, the professor tutored the intermediary in computer science topics such as data models, data structures, and fundamentals of data interpretation. Initially these topics were novel to the intermediary and were intrinsically complicated; however, the professor conveyed these in a clear and straightforward manner and spent time on specific questions that arose.

The intermediary had difficulty with understanding language of computer science. As a result, he frequently had to ask the team for clarification. Overcoming this challenge was simple -- both sides learned to collaborate as a team instead of separating the sciences.

SMILES notation, the conversion of 3-D organic molecules into a character string, is a fundamental concept that the student team was required to understand for the project's success. The research scientists prepared a presentation for the team on SMILES notation in addition to end-group notation. After learning both notations, the intermediary realized that his knowledge was useful for the team. For example, chemical end-groups represent a functional group at the end of a molecule. The intermediary was able to explain end-groups to the initial team, but required the help of computer science students to translate end-groups into a data model. Over time, the intermediary's understanding of computer science concepts increased; this helped him assist the rest of the initial team to understand the application domain.

In addition, the student team used a number of software engineering tools and practices: the SCRUM project management methodology [x], Doxygen [xi], code reviews, and Mob Programming [xii]. The research scientists were not familiar with the SCRUM project management methodology; however, they understood that the student team needed some structure to manage their project tasks. The research scientists' opinion is that using some management methodology is better than using none. SCRUM was chosen because the student team had experience with the methodology during a prerequisite course.

B. Difficulty defining data-models

Chemists and computer scientists communicate in two different ways. Chemists talk in terms of the application domain, while the computer scientists talk in terms of the implementation domain. We needed to bridge that gap by creating a common vocabulary for the team.

We found that one solution to this problem was to define terms and concepts in an easy-to-understand language. For example, the student team selected object role modeling (ORM, Figure 4) to represent a chemical structure as data [xiii]. ORM verbalizes objects and their relationships by using natural language. Defining a data model (or schema) usually involves techniques that optimize queries while maintaining data integrity. A key challenge in developing the schema originated from understanding various relationships and constraints between the polymer structures.

The team began by identifying the core chemical structures and their relationships and then building an ORM data model. The advantage of creating the data model using ORM was to remove the constraints of a particular database implementation so that the team could simply concentrate on higher-level relationships. For example, the team was required to define a constraint that enforced intransitivity between multiple repeat units (a collection of molecular segments, which repeat within a polymer). This relationship is trivial to verbalize (ORM, Figure 5); however, expressing this constraint, and identifying other similar constraints as the schema evolved, presented more challenges.

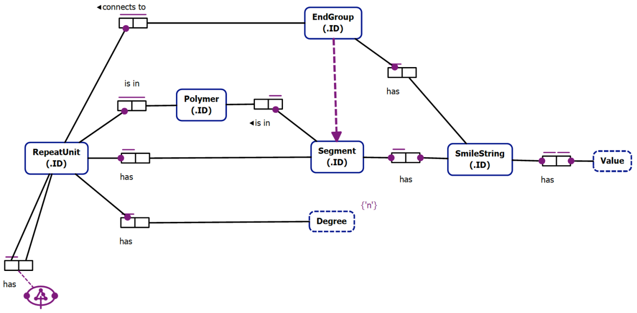
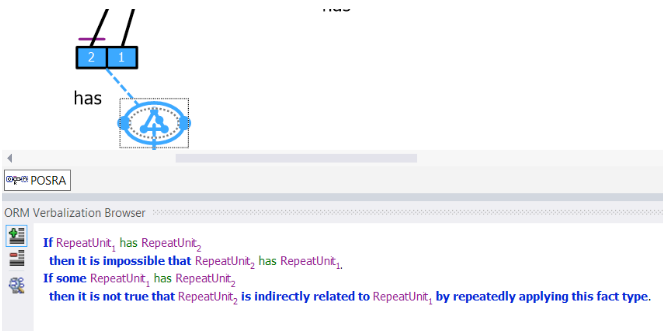


Fig. 4. Object Relational Model



If RepeatUnit$_1$ has RepeatUnit$_2$
then it is impossible that RepeatUnit$_2$ has RepeatUnit$_1$.
If some RepeatUnit$_1$ has RepeatUnit$_2$
then it is not true that RepeatUnit$_2$ **is indirectly related to** RepeatUnit$_1$ **by repeatedly applying this fact type**.

Fig. 5. ORM text output

By expressing these constraints and analyzing the requirements in an abstract way, and as they relate to chemistry, the team overcame some of the challenges with understanding the considerable amount of domain knowledge that needed to be captured in these data structures. Using ORM allowed the team to understand the fundamental database constraints and relationships while defining a platform independent, abstract data model. In addition, by analyzing the key constraints and using ORM to verbalize the data model, in a manner that was consistent with the chemistry, the team gained additional insights into the solution architecture.

C. *Team members not co-located & scheduling conflicts*

Cross-disciplinary teams are usually located in different departments across a single campus rather than in different geographic sites. Our student team has laboratory space located in the Jack Baskin Engineering building on the UCSC campus. Our research team is located at the IBM Almaden Research Center, 35 miles from the UCSC campus. We bridge the location gap using in-person meetings twice a month and online virtual meetings every week we do not get together in person. Moreover, we are able to share information via the central online repository, GitHub [xiv]. This site stores the team's code and documents necessary for the SCRUM process. Fortunately, everyone in the team was familiar with the repository, and could begin contributing from the start of the project.

To deal with scheduling conflicts, the student team allocated time to work together as a group, and used this time for code reviews and implementation sessions. During the code reviews, the team used Doxygen, and manually parsed through the most critical functions of OSRA. In the implementation sessions, the student team worked together in one room, using one computer, while working on one problem. This practice, called Mob Programming [xii], was introduced to the students in the capstone course in a talk given by Woody Zuill, an industry leader that uses this practice with his software development team. It proved to be another way for the student team to assist each other to learn the fundamentals of the application and implementation domains (Figure 6).

Fig. 6. Student team during mob programming

III. Conclusion

We believe we have successfully built a cross-disciplinary team. We believe we have used software engineering tools and techniques, modern technology for virtual meetings, as well as interpersonal techniques to approach and solve some of the challenges we have experienced in building our cross-disciplinary team. We will not know about the real success of our cross-disciplinary team until we have a running prototype to extend OSRA to extract polymer structures from text. Currently, we expect to have code for vertical bracket detection by the end of the capstone course. Our progress gives us confidence that we will have the horizontal bracket detection feature implemented as well.

Acknowledgment

The student team of this paper thanks our advisors, Professor Linda Werner, and Dr.s Julia Rice, Hans Horn, and Amanda Engler, for their support, patience, and attention. They continue to help guide the team through its efforts toward accelerating the process of finding new materials. The authors thank Dr. Igor Filippov [1] for helpful discusssions.

References

[1] I. V. Filippov, "OSRA: Optical Structure Recognition Application", <http://cactus.nci.nih.gov/osra/>, N.p., n.d. Sept. 12, 2009.

[2] I. V. Filippov, M. C. Nicklaus,
"Optical Structure Recognition Software To Recover Chemical Information: OSRA, An Open Source Solution"
J. Chem. Inf. Model., 2009, *49*, 740

[3] A. P. Thurow, C. W. Abdalla, J. Younglove-Webb, B. Gray, "The Dynamics of Multidisciplinary Research Teams in Academia." *The Review of Higher Education"* 1999, 22, 425

[4] "Chapter 1: Introduction to Organic Structures." *UC Davis ChemWiki*. <http://chemwiki.ucdavis.edu/Organic_Chemistry/Organic_Chemistry_With_a_Biological_Emphasis/Chapter__1%3A_Chapter_1%3A_Introduction_to_organic_structure_and_bonding_I> N.p., n.d. Feb. 28. 2014.

[5] D. Weininger. "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules". *J. Chem. Inf. Model,* 1988*,* 28, 31

[6] D. Weininger, A. Weininger, J. L. Weininger, "SMILES. 2. Algorithm for generation of unique SMILES notation". *J. Chem. Inf. Model,* 1989, 29, 97

[7] D. Weininger. "SMILES. 3. DEPICT. Graphical depiction of chemical structures". *J. Chem. Inf. Model,*1990, 30, 237

[8] A. Dalby , J. G. Nourse , W. D. Hounshell , A. K. I. Gushurst , D. L. Grier, B. A. Leland, J. Laufer, "Description of several chemical structure file formats used by computer programs developed at Molecular Design Limited"**,** *J. Chem. Inf. Comput. Sci.* 1992, 32, 244

[9] M. P. Robillard, W. Coelho, G. C. Murphy. "How Effective Developers Investigate Source Code: An Exploratory Study." *IEEE Transactions on Software Engineering* , 2004, 30, 889

[10] J. Sutherland; K. Schwaber. "The Scrum Guide". <Scrum.org>. N.p., n.d. July 2013

[11] D. van Heesch. "Doxygen." <*www.doxygen.org*>. N.p., n.d. Feb. 24. 2014.

[12] W. Zuill. "Mob Programming." <*MobProgramming.org*>. N.p., n.d. Feb. 24. 2014.

[13] T. Halpin, "Object Role Modeling." <**www.orm.net**> N.p., n.d. Feb. 28 2014.

[14] Github <https://github.com/> Feb. 28. 2014.