Building Cyber Infrastructure for Geochronology: A Case Study in Collaborative Software Engineering Research

James F. Bowring
Department of Computer Science
College of Charleston
Charleston, South Carolina, U.S.A.
BowringJ@cofc.edu

ABSTRACT

The Cyber Infrastructure Research and Development Lab for the Earth Sciences (CIRDLES) collaboratively integrates domain-specific software engineering with the efforts of two NSF-supported initiatives in geochronology. Geochronology is the science of determining the age of geological formations. The EARTHTIME initiative pursues consensus-based approaches to geochemical data reduction, and the EarthChem initiative pursues the creation of data repositories for all geochemical data. CIRDLES provides software engineering infrastructure to support the development of software and systems that serve as the cyber infrastructure for geochronology. This collaboration benefits the earth sciences by enabling geochemists to focus on their specialties using robust software that produces reliable results. This collaboration benefits software engineering by providing research opportunities to improve process methodologies used in the design and implementation of domain-specific solutions. Finally, this collaboration benefits the teaching efforts of both software engineers and geochemists by producing tangible, open-source artifacts for students to use. CIRDLES is an example of how research in collaborative software engineering can expand to include collaboration with other sciences and domains.

Categories and Subject Descriptors

K.6.3 [**Management of Computing and Information Systems**]: Software Management—*Software development*; J.2 [**Physical Sciences and Engineering**]: [Earth and atmospheric sciences]

General Terms

Design, Management, Reliability, Standardization

Keywords

Software engineering, Domain-driven design, Collaboration, Cyber infrastructure, Geochemistry

1. INTRODUCTION

Software engineering researchers note that scientists in other academic fields are writing their own software, and the evidence is that these programs could benefit from the use of software engineering science and expertise (e.g., [4, 2, 6].) For example, Kelly argues that the "domain-independent solutions" produced by software engineers have served to divide software engineers from scientists who do computing by a "software chasm [2]."

To address this problem, this paper presents a case study of collaborative software engineering conducted with and for another scientific discipline - earth and atmospheric sciences. This collaboration provides an opportunity for software engineers at the College of Charleston to introduce and deploy software engineering methods and processes for the benefit of another discipline. In addition, this collaboration provides a research platform to evaluate and extend these methods and processes, from design to testing, and to develop a framework for establishing a common language for collaboration with another domain.

The Cyber Infrastructure Research and Development Lab for the Earth Sciences (CIRDLES) is a National Science Foundation(NSF) - sponsored effort. The initial focus is the development of cyber infrastructure for the domain of *geochronology* - the science of determining the age of geological formations. CIRDLES collaboratively integrates domain-specific software engineering with the efforts of two other NSF - supported initiatives: EARTHTIME[1] and EarthChem,[2] which respectively pursue consensus-based approaches to geochemical data reduction, and the creation of data repositories for all geochemical data.

The CIRDLES project has three main goals:

1. to create and implement automated, standardized, and open analysis techniques for geochemistry
2. to seamlessly federate the results of these techniques from many geo-chemistry labs into a shared database
3. to design and build templates for cyber infrastructure that demonstrate this end-to-end system, from mass spectrometer to publication to archiving to retrieval, which will serve as models for subsequent efforts involving all kinds of geochronological data.

[1]http://www.earth-time.org
[2]http://www.earthchem.org

These goals are designed to foster the creation of a cyber infrastructure consisting of software and data structures for geochronology that supports the universal identification of geological samples, transparent data reduction algorithms, robust data formats, Internet-based storage and retrieval, and transparent compilation and aggregation techniques for analysis of records retrieved from the database.

In reaching these goals, CIRDLES is deploying an infrastructure for software engineering that extends beyond the conventional notions of infrastructure to include a common language with which software engineers and geochemists can productively communicate and collaborate. Additionally, CIRDLES is creating infrastructure for the conduct of earth science. Together, these two meanings for infrastructure can inform a broad approach to software engineering research.

The main contributions of this paper are:

1. Description of CIRDLES, an inter-disciplinary collaborative software engineering project; and
2. Description of novel approaches to software engineering infrastructure to support domain-specific development.

2. BACKGROUND

The genesis of this project dates to 1993 in discussions between the author and Sam Bowring of MIT, PI of the Isotope Lab[3] of the Department of Earth, Atmospheric, and Planetary Sciences. The Isotope Lab and the community-supported EARTHTIME initiative are interested in high-precision calibration of earth history. The science of geochronology is computationally intensive because large volumes of raw data produced by mass spectrometers have to be aggregated and reduced using statistical techniques to produce precise geological dates.

According to Bowring, in 1993 and even today, geochemists are dependent on data-reduction software over which they have little control. This software ranges from proprietary products that ship with mass spectrometers to customized Excel spreadsheets to MATLAB routines shared by the community. These specialized applications were not always developed with rigorous processes and hence often have little documentation or repeatable testing. The general effect is that of a black box that everyone trusts.

Bowring's vision was to enlist computer scientists in an interdispciplinary effort to create data-reduction software that was open and understandable and that served the needs of the geochemistry community. His frustration was the lack of knowledge about proper software development in the earth sciences community. The author and he collaborated for fifteen years to explore the possibilities and to produce two programs that began to reify this vision. This collaboration exposed the inherent difficulties in communicating between experts in geochemistry and experts in software engineering. These difficulties range from mundane conflicts in the definitions of words such as *publish* or *error* to more serious misunderstandings about process logic.

[3]http://eaps.mit.edu/research/group/IGLab/

In the meantime, the National Science Foundation began to push for cyber infrastructure development and funded both the EARTHTIME and EarthChem projects. In 1996, Bowring proposed a software program, which eventually came to be known as *U-Pb_Redux*, that would be the heart of an end-to-end data reduction and archiving system. In March 2007, EarthChem held a workshop at the University of Kansas to lay the groundwork for this effort, and later that year provided subsidiary funding for CIRDLES to produce a first generation system during 2008.

3. CIRDLES INFRASTRUCTURE

The Cyber Infrastructure Research and Development Lab for the Earth Sciences (CIRDLES) is the software engineering component of a collaboration that integrates with EARTHTIME and EarthChem. The EARTHTIME initiative is a community supported network of geologists and geochemists who are focused on providing the tools necessary for high precision sequencing of earth history using an integration of radioisotopic dating and stratigraphy. With community support, EARTHTIME is unifying the different approaches to data acquisition and processing. At the same time, the EarthChem project is designing and implementing data repositories for all geochemical data, including radioisotopic geochronological data. CIRDLES provides the crucial link between data production and data archiving. This is apparently the first time this has been attempted in the Earth sciences, and will serve as a template for all other radioisotopic systems such as 40Ar/39Ar, U-Th-He, and Lu-Hf.

CIRDLES is tasked with producing an open-source software program named *U-Pb_Redux* and the associated data structures that will work in concert as the heart of the proposed cyber infrastructure. To accomplish this task requires the building of software engineering infrastructure to support this collaboration with geochemistry and to support related research efforts. This section presents that infrastructure and its development.

3.1 Establishing a common language

Collaboration across disciplines or domains requires the establishment of a common language. This intuitive fact has now been formally proposed by Evans [1]. This effort by CIRDLES found that a useful approach to establishing a common language was the creation and use of data dictionaries. A *data dictionary* codifies data by providing specific names and definitions for both data elements and data processes. In the case of geochemistry, most of the data elements are isotopic ratios, their associated uncertainties, and derived ratios, dates, and their uncertainties. CIRDLES identified the key properties of entries in the data dictionary as follows:

1. a listing of all inputs, intermediate results, partial derivatives, terms and mathematical expressions used at every step of the data reduction
2. a naming convention for this list that supports both geochemistry and variable naming constraints of modern computer languages (Java, C#)
3. a specification of the types of uncertainty (percent or absolute) for every numerical data item in the calculations and in the archival database

4. a specification of the precision of every numerical data item
5. a specification for the organization of the data in the archival database
6. a listing of the name and properties of every data item required in the archival database
7. specification of Extensible Markup Language (XML) Schema Definitions (XSD) to provide an open and public structure for data transmission as XML files at each step of the proposed processes
8. specification of programs in Extensible Stylesheet Language Transformations language (XSLT) to perform automated translations of the XML data into appropriate text, spreadsheet, graphical, etc. representations

This approach of specifying the common language in advance of developing the software solutions is the first step in creating the infrastructure for this project and represents a meta-use of software engineering process management as specified in many textbooks, e.g. Summerville [5].

3.2 Adopting a process model

The very notion of a software development process model was foreign to geochemists so CIRDLES engineers proceeded with a generic iterative approach that utilized prototypes. Prototypes became a powerful communication tool because our collaboration even with a common language often stumbled on unpredicted misunderstandings and misconceptions that could often be resolved with these exploratory prototypes. Other researchers also show the usefulness of prototypes in advancing the pace of development. For example, MacCormack and colleagues find that prototypes are a significant factor in predicting productivity when used in conjunction with a coherent set of development practices [3].

As part of specifying the process model, CIRDLES engineers introduced the notion of version control for all documents to the geochemists in the form of Subversion.[4] Anecdotally, the author reports that, once exposed to version control, the geochemists became energetic adopters. CIRDLES also adopted the use of a web site as an interface to key information, a blog for discussions, and a wiki for progress reports and access to software and documentation.

3.3 Cyber infrastructure requirements

The product of this collaboration, as stated above, is cyber infrastructure and participants took pains to enumerate its key non-functional requirements early on as part of our learning about each other's domain. The key requirements were:

1. design and implement *U-Pb_Redux* as an open-source and platform-independent system
2. provide that the data reduction and analysis equations and algorithms are transparent and documented
3. involve undergraduate and graduate students in the development process
4. include pedagogical tools such as visualizations of process to aid in teaching geochemistry and software engineering
5. provide documentation in the form of specifications, references, derivations, user manuals, and help engines
6. provide various types of interfaces to support both human and automated interaction with the common archival database
7. provide for the sharing and aggregation of schema-compliant data files between individual users to support collaboration within and between labs
8. provide visualizations of all calculations
9. provide visualizations of the propagation of uncertainties during the data reduction
10. design and implement a full suite of open-source and documented tests for all parts of the system from tests of single computational units to full integration tests
11. develop and provide synthetic data sets for demonstrations of the system and for testing of the cyber infrastructure

This requirements listing for the product came to drive the choices for our collaborative infrastructure by providing an organized vision for the group. This listing also became the basis for iteratively refining our processes.

3.4 Collaborative infrastructure

Table 1 lists the current configuration for our development infrastructure:

Table 1: Infrastructure specifications.

open source license	Apache, Version 2.0[5]
version control	Subversion
programming language	Java
integrated development environment	NetBeans[6]
testing environment	JUnit
help engine	Java Help
documentation	Java docs and various
UML environment	Enterprise Architect[7]
transmitted data structure	XML
Wiki	PmWiki[8]
Blog	WordPress[9]
distribution	Subversion
bug reporting	email
undergraduate researchers from	College of Charleston[10]
graduate researchers from	MIT[11]

This infrastructure is evolving as the collaboration matures and participants intend to make the infrastructure, the rationalizations and experiences available to others as part of the published cyber infrastructure.

[4] http://subversion.tigris.org/

[5] http://www.apache.org/licenses/LICENSE-2.0

[6] www.netbeans.org

[7] http://www.sparxsystems.com.au

[8] http://pmwiki.com

[9] http://wordpress.org

[10] www.cs.cofc.edu

[11] http://eapsweb.mit.edu/

3.5 Cyber infrastructure produced

CIRDLES will formally release the first version of the *U-Pb_Redux* system at the October 2008 annual meeting of the Geological Society of America (GSA). Interested readers can monitor progress and learn more about the project from the CIRDLES website: https://cirdles.cs.cofc.edu.

4. RELATED WORK

Diane F. Kelly argues that there is a software chasm between software engineering and scientific computing that dates from the 1960s [2]. According to Kelly, the chasm developed because software engineering research has until recently focused on "domain-independent" approaches. One effect of these approaches was the "isolation of the scientific-computing community" from the software-engineering community because the former needed domain-specific solutions and the latter was not interested in providing them. Sanders and Kelly look more closely at this chasm and survey 16 scientists from 10 disciplines to gain insights into the practice of scientific computing [4]. They focused their analysis on risk management and discovered that there exists a systemic weakness in the rigorous testing of software. According to the authors, this weakness stems from a shortage of formal testing methods designed specifically for for scientific application software. Wilson argues that there is a "bottleneck in scientific computing" that he attributes to "computational illiteracy [6]." He reports anecdotally that scientific software is often produced without any basic knowledge of software engineering and then used uncritically to produce reportable results.

These works illustrate the need for a new focus in collaborative software engineering research that will address the needs of domain-specific development for the sciences. CIRDLES represents a first step in this direction by creating a collaboration to produce domain-specific software that is yielding insights into how to manage the attendant processes and risks.

5. CONCLUSIONS AND FUTURE WORK

This paper presented the Cyber Infrastructure Research and Development Lab for the Earth Sciences (CIRDLES) as an example of inter-disciplinary collaborative software engineering. This paper has described the ongoing efforts to establish a common language and workable development processes that together enable a successful collaboration.

This project is not completed and CIRDLES has a number of ongoing as well as future research questions to pursue. First, participants intend to incorporate into the software, per the requirements in Section 3.3, an array of interactive visualizations that illustrate the mechanics of the data reduction as well as the propagation of uncertainties. The research question involves inventing compelling visualizations that teachers and students alike will use to enhance their understanding. The solutions likely reside in a confluence of computer science, earth sciences, and visual arts. Second, as the software is released to the geochemistry community, CIRDLES will need to enhance the infrastructure to accommodate the inevitable resultant feedback. While such mechanisms exist in the software engineering community, they do not exist for this interdisciplinary effort. Third, to leverage this cyber infrastructure for use in parallel efforts in the earth sciences, CIRDLES will explore ways to abstract the basic mechanisms and software engineering infrastructure to provide a body of knowledge. Fourth, participants intend to advance the development of formal testing methods geared to testing software created for and by the scientific community. Finally, the author believes that much work remains to be done in discovering how best to facilitate collaborations between software engineers and scientists in other domains.

6. ACKNOWLEDGMENTS

This work is supported by the Department of Computer Science at the College of Charleston, by EARTHTIME and by EarthChem. Funding is provided by National Science Foundation Grant No. EAR-0522222. I thank Chris Starr of the College of Charleston, Sam Bowring of MIT, and Doug Walker of the University of Kansas for their support and collaboration.

7. REFERENCES

[1] E. Evans. *Domain-Driven Design: Tackling complexity in the heart of software*. Addison-Wesley, Upper Saddle River, NJ, 2004.

[2] D. F. Kelly. A software chasm: Software engineering and scientific computing. *IEEE Software*, 24(6):120–119, 2007.

[3] A. MacCormack, C. F. Kemerer, M. Cusumano, and B. Crandall. Trade-offs between productivity and quality in selecting software development practices. *IEEE Software*, 0740-7459(03):78–86, 2003.

[4] R. Sanders and D. F. Kelly. Dealing with risk in scientific software development. *IEEE Software*, 25(4):21–28, 2008.

[5] I. Sommerville. *Software Engineering*. Pearson Education Limited, London, 2007.

[6] G. Wilson. Where's the real bottleneck in scientific computing? *American Scientist*, 94(1):5–8, 2006.