Self-evaluating, Collaborative Labs for Formal Sciences

Karl Lieberherr
Northeastern University
CCIS
Boston
lieber@ccs.neu.edu

ABSTRACT

The Scientific Method is widely used but there is no generic cyberinfrastructure that works across scientific domains and that enforces the rules of a refutation-based scientific community. In the Scientific Method one formulates claims in a form that could conceivably be falsified by a test using a protocol. The refutation protocol is part of the claim which makes it explicit when refutation is successful.

We propose to provide a generic cyberinfrastructure for the Scientific Method which can be customized for numerous scientific domains. The benefits are (1) a uniform interface across scientific domains (2) faster and targeted scientific progress because claims are exposed to more refutation attempts. They are in refutable form on the web. (3) the cyberinfrastructure adapts to the skill level of the scholars (players).

While Google's mission is to "organize the world's information and make it universally accessible and useful" our mission is "to implement cyberinfrastructure to develop and support constructive, domain-specific knowledge."

RESEARCH GOALS

The Scientific Method has not yet been implemented as a generic software tool that can be customized for various domains to develop new knowledge. We present our design of such a generic tool and propose how to improve the current implementation. Applications range from research to teaching (MOOC) courses in STEM areas.

In this proposal we focus on formal sciences although the idea applies to both formal sciences and other sciences. Unlike other sciences, the formal sciences are not concerned with the validity of claims based on observations in the real world, but instead with the properties of formal systems based on definitions and rules. Examples of formal sciences are: logic, mathematics, theoretical computer science, information theory, systems theory, decision theory, statistics.

Submitted for review to Google. Previous work supported by GMO and Novartis.

Our goal is to develop a cyberinfrastructure to define labs that are used to develop or disseminate constructive, domain-specific knowledge. Think of a lab as a virtual world where scholars make claims and other scholars refute, strengthen or agree with those claims. The labs are inhabited either by human scholars or by avatars.

In 3 pages we cannot completely describe the Scientific Community Game. Therefore we have written a separate 8 page description of the game [8].

We have developed the Scientific Community Game (SCG) over the last few years as a model for the Scientific Method. SCG organizes the development of true and optimal claims about domain-specific, constructive knowledge. While SCG itself is a formal system with formal properties which is the result of a mechanism design problem, it has a social engineering motivation.

Problem Statement: Lab Design

Defining good labs is a challenging task. Lab design is like program design but the program is executed by machines and people. A lab implementation consists of the implementation of a few method signatures (such as valid, quality, belongsTo, etc.) which define a sociotechnical system in which scholars solve the problem posed by the lab design. In modern terminology, lab design is a part of programming the human brain [3].

We propose to solve the following problems related to lab design. (1) dynamic checking of lab designs. (2) dividing scholars into pairs to optimize the innovation/teaching effect. (3) distributing credit for partial results. (4) claim reductions.

This new software will be integrated with our current Scientific Community Game (SCG) software [1]. If time permits, we would like to port the system to the cloud to make it flexible with respect to resource demands.

DESCRIPTION OF WORK

We refer to the numerated list from above.

1. We develop a dynamic checking tool that gives useful feedback on the quality of a lab before it is given to human scholars or avatars.

   The idea behind the dynamic checking tool is that the lab definition defines a baby avatar that properly follows the rules of the lab but which has very poor intelligence. The baby avatar implements a few methods, including: propose, refute, strengthen and agree.

The parameterized nature of SCG is both powerful but it also makes it challenging to use SCG. The generic SCG software is parameterized by a lab involving sets Instance, Solution, Claim and functions valid, quality, stronger along with configuration information such as the maximum sizes of objects.

The goal of this proposal is to simplify the testing of labs. I have designed dozens of labs and used them in my courses. A small mistake in a lab definition can make it useless. Our current way of testing a lab is to generate two baby avatars and let them compete against each other in a binary game. The baby avatars make use of randomization for both their exploratory and performatory actions. If the binary game works, we know that the basic mechanisms work but we don't know about loopholes that can creep in. We propose to use dynamic contract checking to help debugging of the labs. The lab designers will have to write pre and post conditions for the methods that make up a lab definition.

2. When pairing scholars for binary games, it is important that one of the scholars has good prerequisites. This is important for teaching but also for innovation. The pairing will be done based on previous performance and by filling out a questionnaire.

3. To keep the scholars motivate it is important that they get partial credit for good partial results. For example, when Alice makes a claim that later gets strengthened by Bob, Alice should get partial credit.

4. Reductions between claims are important in practice to find successful refutations. Reductions are of the form: if you can refute claim C1 in lab L1, then you can refute claim t(C1) in lab t(L1) for a mapping t. Examples of such reductions are in [11, 10]. We propose to extend SCG to directly address the expression of reductions. Looking for such reductions is a successful problem solving technique.

In addition there several other features that we would like to add to SCG. But they are probably beyond the scope of this proposal: support for long-running labs, tools to mine the game histories for information about how to improve the avatars, tools that take the best features of avatars and combine them into new and better avatars, using machine learning to help with avatar development, etc.

Availability of results

Our current software is already available one the web [1] and the new versions supported through Google will continue to do so. Beyond publishing papers in conferences and journals, we will support the academic community by providing a new model for publishing claims.

Use of previous work at Google

I am the initiator of the Demeter project which proposed the Law of Demeter (LoD). John Lamping sent me an email about a year ago, mentioning that Google makes active use of LoD. The Clean Code Talks series in the Google Tech Talks refers to the Law of Demeter as a promoter of clean code (Miskov Hevery, Google). This search returns about a dozen posts by Google referring to the LoD:

```
http://www.google.com/
search?q=demeter+site%3Agoogletesting.
blogspot.com
```

Budget

The PhD student will contribute to the intellectual development of the dynamic lab testing tool and will improve the current code base. The goal is to get a tool on the web that makes the global brain more productive.

RELATED WORK

SCG has not grown in a vacuum. We make connections to several related areas.

Crowd Sourcing and Human Computation

There are several websites that organize competitions. What is common to many of those competitions? We believe that SCG provides a foundation to websites such as TopCoder [15] or the more specialized kaggle.com.

SCG makes a specific, but incomplete proposal of a programming interface to work with the global brain [3]. What is currently missing is a payment mechanism for the scholars and an algorithm to split workers into pairs based on their background.

SCG is a generic version of the "Beat the Machine" approach for improving the performance of machine learning systems [2].

Scientific discovery games, such as FoldIt and EteRNA, are variants of SCG. [4] describes the challenges behind developing scientific discovery games.

Logic and Imperfect Information Games

Logic and Games: finding a proof for a theorem = finding a defense strategy for a claim.

Logical Games [13], [5] have a long history. SCG builds on Paul Lorenzen's dialogical games [6].

Foundations of Digital Games

According to Jonas Linderoth [12] games challenge two aspects of human nature: our ability to choose appropriate actions and our ability to perform appropriate actions. [12] views gaming as a cycle between interrelated exploratory and performatory actions.

What are the exploratory and performatory actions in SCG? Exploratory actions are: (1) proposing a claim which means choosing from a set of claims. (2) choosing an action: refute, agree or strengthen a given claim. Performatory actions are: (1) the proponent should defend the proposed claim, (2) a claim for which the refute action was chosen should refute the claim, etc. In SCG we also have a cycle of interrelated exploratory and performatory actions.

Architecting Socio-Technical Ecosystems

This area has been studied by James Herbsleb and the Center on Architecting Socio-Technical Ecosystems (COASTE) at CMU http://www.coaste.org/. A socio-technical ecosystem supports straightforward integration of contributions from many participants and allows easy configuration.

SCG has this property and provides a specific architecture for building knowledge bases in (formal) sciences. Collaboration between scholars is achieved through the scientific discourse which exchanges instances and solutions. The structure of those instances and solutions gives hints about the solution approach. An interesting question is why this indirect communication approach works so well.

Online Judges

An online judge is an online system to test programs in programming contests. A recent entry is [14] where private inputs are used to test the programs. [15] includes an online judge capability but where the inputs are provided by competitors. This dynamic benchmark capability is also expressible with SCG: The claims say that for a given program that all inputs create the correct output. A refutation is an input which creates the wrong result.

Origins of the Scientific Community Game

A preliminary definition of SCG was given in a keynote paper [9]. [7] gives further information on the Scientific Community Game. The original motivation for SCG came from the two papers with Ernst Specker: [10] and the follow-on paper [11]. Renaissance competitions: the public problem solving duel between Fior and Tartaglia, about 1535, can easily be expressed with the SCG protocol language.

SUMMARY

The SCG project covers a niche that has been explored by a few other projects, but not in a general way. We are the first to take a Scientific Method approach to competitions that has broad applicability.

REFERENCES

1. A. Abdelmeged and K. J. Lieberherr. SCG Court: Generator of teaching/innovation labs. Website, 2011. `http://sourceforge.net/p/generic-scg/code-0/110/tree/GenericSCG/` .

2. J. Attenberg, P. Ipeirotis, and F. Provost. Beat the machine: Challenging workers to find the unknown unknowns. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

3. A. Bernstein, M. Klein, and T. W. Malone. Programming the global brain. *Commun. ACM*, 55(5):41–43, May 2012.

4. S. Cooper, A. Treuille, J. Barbero, A. Leaver-Fay, K. Tuite, F. Khatib, A. C. Snyder, M. Beenen, D. Salesin, D. Baker, and Z. Popović. The challenge of designing scientific discovery games. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, FDG '10, pages 40–47, New York, NY, USA, 2010. ACM.

5. W. Hodges. Logic and games. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Spring 2009 edition, 2009.

6. L. Keiff. Dialogical logic. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Summer 2011 edition, 2011.

7. K. Lieberherr. The Scientific Community Game. Website, 2009. `http://www.ccs.neu.edu/home/lieber/evergreen/specker/scg-home.html`.

8. K. J. Lieberherr and A. Abdelmeged. The Scientific Community Game. In *CCIS Technical Report NU-CCIS-2012-19*, October 2012. `http://www.ccs.neu.edu/home/lieber/papers/SCG-definition/SCG-definition-NU-CCIS-2012.pdf`.

9. K. J. Lieberherr, A. Abdelmeged, and B. Chadwick. The Specker Challenge Game for Education and Innovation in Constructive Domains. In *Keynote paper at Bionetics 2010, Cambridge, MA, and CCIS Technical Report NU-CCIS-2010-19*, December 2010. `http://www.ccs.neu.edu/home/lieber/evergreen/specker/paper/bionetics-2010.pdf` .

10. K. J. Lieberherr and E. Specker. Complexity of Partial Satisfaction. *Journal of the ACM*, 28(2):411–421, 1981. `http://www.ccs.neu.edu/home/lieber/p-optimal/JACM1981.pdf` .

11. K. J. Lieberherr and E. Specker. Complexity of Partial Satisfaction II. *Elemente der Mathematik*, 67(3):134–150, 2012. `http://www.ccs.neu.edu/home/lieber/p-optimal/partial-sat-II/Partial-SAT2.pdf`.

12. J. Linderoth. Why gamers don't learn more: An ecological approach to games as learning environments. In L. Petri, T. A. Mette, V. Harko, and W. Annika, editors, *Proceedings of DiGRA Nordic 2010: Experiencing Games: Games, Play, and Players*, Stockholm, January 2010. University of Stockholm.

13. M. Marion. Why Play Logical Games. Website, 2009. `http://www.philomath.uqam.ca/doc/LogicalGames.pdf`.

14. J. Petit, O. Giménez, and S. Roura. Jutge.org: an educational programming judge. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, SIGCSE '12, pages 445–450, New York, NY, USA, 2012. ACM.

15. TopCoder. The TopCoder Community. Website, 2009. `http://www.topcoder.com/`.

Karl Lieberherr

A. Academic and Professional History

POSITION: Professor, since 1985
ADDRESS: College of Computer Science, Northeastern University, Boston MA 02115.
EMAIL AND PHONE: lieber@ccs.neu.edu; (617) 373-2077
EDUCATION: Dipl. math. ETH, 1973, ETH Zurich, Switzerland
Dr. sc. math. ETH, 1977, ETH Zurich, Switzerland

POSITIONS HELD
1979-1983, Assistant Professor, EECS, Princeton University
1983-1985, GTE Laboratories, Inc., M.T.S. and Principal M.T.S.

RESEARCH ACCOMPLISHMENTS
Theory of P-Optimal Algorithms for Constraint Satisfaction, Non-chronological backtracking based on clause learning, Hardware Description Language Zeus, Demeter project (started in 1985), Law of Demeter, early contributor to Adaptive and Aspect-Oriented Programming.

KEYNOTE SPEAKER
International Conference on Software Engineering 2004, Bionetics 2010.

B. Selected Publications

1. Karl Lieberherr and Ernst Specker, "Complexity of Partial Satisfaction II", *Elemente der Mathematik*, vol. 67, no. 3, 2012, pages 134-150, European Mathematical Society Publishing House.

2. Mira Mezini and Karl Lieberherr, "Adaptive Plug-and-Play Components for Evolutionary Software Development" *1998 Proc. Object-Oriented Programming Systems, Languages and Applications Conference (OOPSLA), Special Issue of SIGPLAN Notices*, Vol. 33, no. 10, pages 97-116, ACM Press.

3. Karl Lieberherr and Boaz Patt-Shamir and Doug Orleans, "Traversals of object structures: Specification and Efficient Implementation", *ACM Trans. Program. Lang. Syst.*, vol. 26 (2), 2004, pages 370–412, ACM Press.

4. Karl Lieberherr, "Adaptive Object-Oriented Software: The Demeter Method with Propagation Patterns", *PWS Publishing Company*, 1996, 650 pages, ISBN 0-534-94602-X.

5. Karl J. Lieberherr, "Controlling the Complexity of Software Designs", International Conference on Software Engineering, 2004, Edinburgh, Scotland, pages 2-11, editors Jacky Estublier and David Rosenblum, ACM Press, invited keynote presentation.

C. Software Projects, Collaborating Institutions, and Individuals

Xerox PARC, IBM Yorktown Hights, Citibank, SAIC. Collaborating individuals include: Mira Mezini (University of Darmstadt), Jens Palsberg (UCLA), Boaz Patt-Shamir (Tel Aviv University).

D. Selected Ph.D. students (and current location):

Crista Lopes, 1997, University of California at Irvine; Linda Seiter, 1996, John Carroll University; Ignacio Silva-Lepe, 1994, IBM T.J. Watson Research Center. Paul Bergstein, 1994, University of Massachusetts at Dartmouth.