SAGE Intelligent Hinting, DevOps, and Publication Proposal

COMS 6901 – Projects in Computer Science, Spring 2018

Alex Dziena / ad3363, Tomer Zwi / tz2247

Abstract

We propose continuing the work done on Intelligent Hinting, DevOps, and Publication of SAGE's feasibility study in previous semesters, in particular building on the progress made in Spring 2018. Specifically, we propose an integration of the existing hinting system, the addition of a natural language (chatbot) hinting interface, DevOps enhancements to improve the stability of the platform for continued field studies in the Fall semester, and reducing the time-to-productivity for future researchers, and contributing a new reference architecture to the publication draft.

We will specifically be addressing the following Epics and Features:

Epic	Feature
Gameful Intelligent Tutoring	Intelligent Hinting MVP
SAGE Integration	DevOps MVP
	Workstream Integration
Survey, Field Study Design, and Publication Strategy	SAGE Feasibility Study & Publication

Introduction

As the number of researchers and workstreams comprising the SAGE project has steadily grown over the course of the past several semesters, and as SAGE prepares for an initial publication and continuing field studies in the fall, a focus on integration of work done by multiple teams, system stability, and time to productivity has become increasingly important. Our focus this semester will be in three primary areas:

Gameful Intelligent Tutoring

Over the course of this semester, we plan to fully integrate the existing intelligent hinting system into the SAGE UI in such a way that future intelligent hinting modules can be easily developed and integrated, and we plan to prove this integration concept by creating a natural language interface for students to "request" hints by asking questions in a chat interface.

SAGE Integration

We plan to improve future researcher time-to-productivity and the systems stability by implementing several DevOps enhancements, including automated configuration management, Newman integration for automated local API testing, improving test coverage of all SAGE codebases, and developing a set of coding standards for the project.

We further plan to fully document SAGE's modified Agile project management methodology, and develop a set of Use Cases and Personas for SAGE. This will improve the onboarding / orientation of future researchers, which should improve time-to-productivity for new researchers in future semesters.

Survey, Field Study Design, and Publication Strategy

At the end of this semester, we aim to have a draft of SAGE's feasibility study ready for submission to ACM Transactions on Computing Education. In particular our team will be focussed on improving the reference architecture proposed in the paper, and evaluating and potentially improving existing or creating new data visualizations.

Related Work

Gameful Intelligent Tutoring

[Hints: Is It Better to Give or Wait to Be Asked?]
Razzaq, et al. found that students learned more reliably when they were provided with a mechanism to receive hints-on-demand rather than being provided with hints proactively in a "just-in-time" hinting system[13]. Because this effect was more pronounced in students who asked for a larger number of hints, the study may suggest that the relative benefit of hints-on-demand vs proactive hints increases as a student needs increasing amounts of assistance. This paper provided us the inspiration and rationale for a natural language interface to an on-demand hinting system.

[Autonomously Generating Hints by Inferring Problem Solving Policies]
Piech, et al. performed analysis on a large amount of student solutions data gathered from Code.org and proposed several different path modeling algorithms used to learn a Problem Solving Policy (PSP), a policy that returns a suggested next partial solution for all partial solutions to a given puzzle, i.e. a PSP π is defined as $s' = \pi(s)$ for all $s \in S$, with S being the set of all possible solutions. The algorithms used to learn this PSP were compared, with a class of algorithms called Desirable Path algorithms (Possion Path and Independent Probable Path) performing the best - producing PSPs closest to the paths contained in an expert-labeled set of suggested paths.[17]

“Poisson Path”-based PSP learning and hint generation were implemented last semester [18]. This semester, we will build on that work by integrating the hint generation engine with SAGE’s UI, and productionizing the policy learning algorithms to allow for near real-time reinforcement learning of policies.

[Delivering Hints in a Dialogue-Based Intelligent Tutoring System]
Zhou, et al present an implementation of CIRCSIM-Tutor, a dialogue-based intelligent tutoring system. The system used a a labeled corpus of dialogue (questions and answers) from expert tutors to develop a set of hinting strategies used to select appropriate hint templates and parameters as responses to classes of questions. CIRCSIM-Tutor does not support online-updating, but is context aware of it’s previous interactions with a student during a single session.[6]
We seek to build on this work by implementing a reinforcement learning algorithm to allow for online learning by the chatbot agent. Possible candidates that we will evaluate include SARSA, temporal difference learning, and Q-learning[19]. Since we do not have a labeled corpus on which to train, we will rely on semi-supervised learning by assigning reward values to the next student move, and updating the policy in an online (or near online, in the case of SARSA) manner.
The CIRCSIM-Tutor paper also provides an evaluation framework for the efficacy of different hinting strategies, and a set of heuristics to select appropriate hint types. We will use this methodology to evaluate the hinting strategies learning by our chatbot, and display results in SAGE’s researcher interface.[6]

SAGE Integration

[Dimensions of DevOps]
Lwakatare, et al. performed a survey of DevOps practices and identified four primary dimensions of DevOps practice: collaboration, automation, measurement, and monitoring. The paper goes on to specify a conceptual framework for characterizing DevOps practices. The paper does not detail concrete practices or provide reference architectures for implementing DevOps practices, but has been useful in conceptualizing DevOps implementation strategies in previous semesters and was used to identify many of the goals for this semester’s DevOps work including configuration management and test coverage measurement.[3]

[Role of collective ownership and coding standards in coordinating expertise in software project teams]
This paper explores expertise coordination as an important emergent process through which software project teams manage software development challenges, in particular within the framework of Extreme Programming (XP). Maruping et al examine the relationship between collective ownership and coding standards with software project technical quality in a field study of 56 software project teams comprising 509 software developers, and found that collective ownership and coding standards play a role in improving software project technical quality. They

find that coding standards strengthens the relationship, resulting in higher technical quality. This work inspired and will guide development of SAGE’s coding standards documentation.[11]

[Agile Approaches on Large Projects in Large Organizations]
This research explores the apparent conflicts between agile and traditional project management in large organizations first implementing agile practices. It also defines several of the roles and practices used in agile project management, highlighting the importance of communication and stakeholder management. [7]
This research has informed our development of a modified agile project management methodology, and forms a partial basis for the project management methodology documentation we will create this semester.

Survey, Field Study Design, and Publication Strategy

[The concept of reference architectures]
Cloutier, et al. examine reference architectures with the goal of providing a more precise definition of their components and purpose. They propose that the value provided by reference architectures lies in the distillation of (in many cases) thousands of person-years of work, a shared baseline for multiple, often cross-functional teams, and guidance for future work. We will reference the architecture evaluation methodologies and structure / component suggestions put forth by this paper while redeveloping SAGE’s reference architecture included in the feasibility study draft.[15]

[A framework for analysis and design of software reference architectures]
This paper provides a tool in the form of an analysis and design framework, for the creation of software reference architectures based on three primary dimensions: context, goals, and design. The paper goes on to define five types of reference architectures into which architectures under analysis can be classified:

1. classical, standardization architectures for use in multiple organizations
2. classical, standardization architectures for use in a single organization
3. classical, facilitation architectures for use in multiple organizations
4. classical, facilitation architectures for use in a single organization
5. preliminary, facilitation architectures for use in multiple organizations

The paper validates the framework by applying it to analysis of 24 reference architectures.
We plan to use the framework proposed by this paper to design and evaluate our updated reference architecture.[16]

[The visual display of quantitative information]
This book is widely used in industry and academia, and provides an exploration and set of recommendations for the design of statistical graphics, charts, tables. It also provides an analysis framework for selecting appropriate data visualizations for a given data set, attempting to optimize for precision, efficacy, and speed of analysis. We will use this book as a reference

when evaluating and potentially redesigning the data visualizations in the feasibility study draft.[12]

Proposal

Gameful Intelligent Tutoring: Intelligent Hinting MVP

Since the existing HMM and Poisson Path for modeling and clustering student learning paths as detailed in Intelligent Hinting 1.1 Final Report [18] already have a basic API for serving hints, we will begin by querying this API from the sage-scratch swf to make hint data available to the UI. Afterwards we will reuse the block suggestion UI engine built in a previous semester to render the hint. Once this basic integration is complete, we will extend it to support textual hints (rendered in the avatar UI), allowing us to build a chatbot engine and interface.

To make the chatbot as responsive as possible, we will extend the existing hinting system to allow for near-continuous updating of student behavior models. This will require building a new API endpoint to allow for model updating, and productionizing the model training pipeline built last semester. This pipeline is built in Python, which will require extending our continuous build system and other DevOps infrastructure to support a new language.

The chatbot responses will utilize the student's current path cluster assignment and inferred student classification (Extreme Movers, Movers, Stoppers, and Tinkers) to select an appropriate hint on demand. We will infer intent from the student's input using multiple methodologies, including n-gram, bag of words, and word2vec, then select the appropriate hint using the intent, student classification, path cluster assignment, and path cluster progression as inputs.

Once integration with the Gameful Affinity Space is complete, we could extend the researcher interface to analyze and display the efficacy of the hinting system for each student classification: Extreme Movers, Movers, Stoppers, and Tinkers. Efficacy scores per student classification could be computed by taking the average reduction in path length, or wall clock time to solve a puzzle per number of hints provided. With on-demand hinting via the chatbot implemented, we could also take number of hints requested as an input. For parson's puzzles, we could also take the student's score into account.

SAGE Integration: DevOps MVP

The goal of the DevOps work this semester is stability of the platform for the Spring semester, and reducing the time-to-productivity for future researchers. We seek to achieve this using configuration management, test automation, increased test coverage, and documentation of coding standards.

Using configuration management will serve two purposes:

a. It will reduce the time-to-productivity for future researchers,
b. and it will allow for faster disaster recovery and migration of the dev and uat shared environments.

Part of this semester's work will include researching and selecting between three class-leading configuration management solutions: Puppet, Ansible, and Chef. We will evaluate these using a principled approach, measuring (at least) these attributes of each system:

a. Maturity
b. Complexity
c. Fault tolerance / Consistency, Availability, and Partition tolerance (CAP theorem)[14]
d. Estimated time to implement
e. Maintenance overhead
f. Interface simplicity
g. Use-case coverage

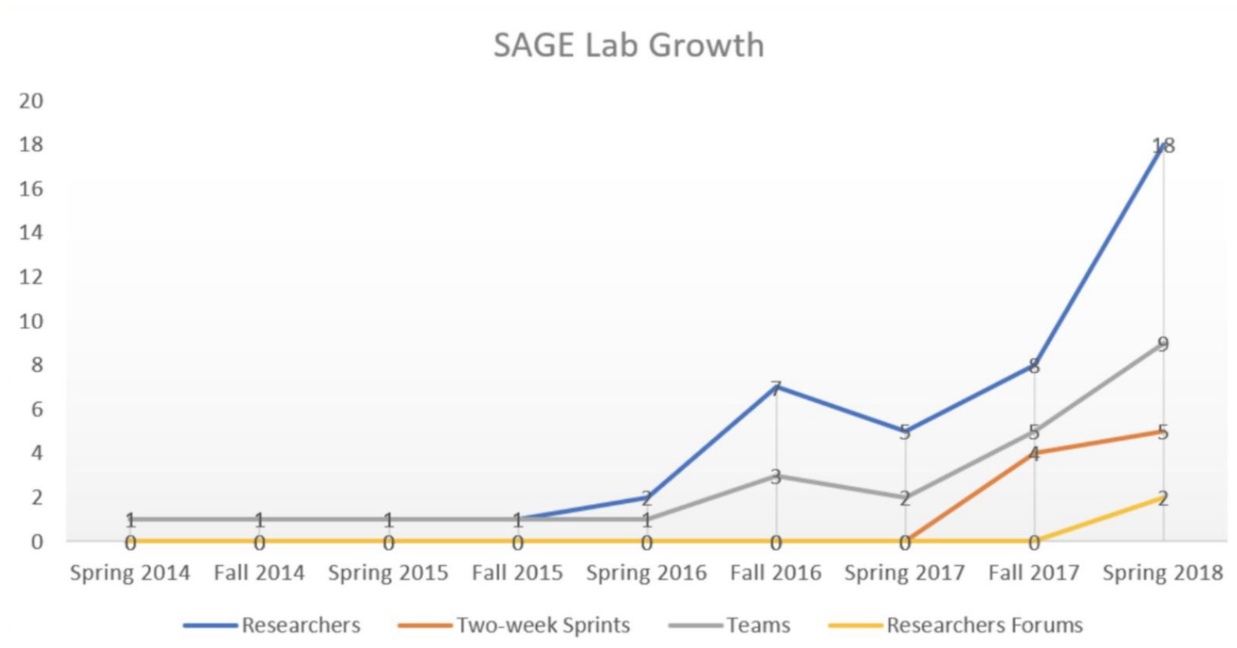
We will integrate Newman to facilitate testing of sage-node and sage-frontend APIs in local development environments, and allow for automated API testing in our continuous integration system. Continuous testing of our APIs will further enhance system stability, and reduce the number of API regressions in future semesters, which will be critical to successful integration of workstreams in the fall and future semesters.

We will improving test coverage for all code bases, including integration of AsUnit into sage-scratch, which will allow for "push-on-green" deployment to shared environments. "Push-on-green" is a methodology in which only the latest build that passes all, or a sufficient proportion of, test cases is pushed to a shared environment.

Adding coding standards documentation has been found to play a role in improving software project technical quality[11]. We believe that adding coding standards to SAGE's documentation will also reduce the time to productivity for new researchers, maintain the health of the SAGE codebase, and allow for easier future integration of linting, mutation, and static analysis tools.

SAGE Integration: Workstream Integration

SAGE has seen a large increase in the number of researchers and disparate workstreams over the past five semesters, with Spring 2018's research group being the largest and most diverse in terms of workstreams.[20] As stated in our previous work on workstream integration, competing priorities, time and resource constraints, and limited attention to process can result in decreased progress toward team goals, and limit the productivity increases that should be associated with team growth, with this problem compounding superlinearly to team size [21].



Our focus in the context of Workstream Integration is optimizing the onboarding and orientation process for future researchers. Use case and persona analysis has been found to be effective in improving usability and user understanding of software systems in a representative case study on a smaller project [10]. We believe that these tools can also improve researcher productivity, and the cohesion of SAGE as it is under development, by providing better context to future researchers. Adding a use case and persona list to SAGE documentation will also help facilitate future field study and publication work.

Documenting and formalizing a project management methodology framework will help with workstream integration in future semesters, provide a framework for future research teams to self-manage, and improve scheduling and collaboration for SAGE publications.

Survey, Field Study Design, and Publication Strategy: SAGE Feasibility Study & Publication

We will focus on developing the reference architecture proposed in SAGE's draft feasibility study to prepare the study for submission to ACM Transactions on Computing Education at the end of the semester. In particular, we will include an abstract API specification for the components of the architecture, and a set of process flows through the architecture for several use cases for students, instructors, and researchers.

We will also evaluate and potentially improve or augment the data visualizations in the draft, with a particular emphasis on data visualization best practices[12].

Timeline

Date	Milestone
2018-06-16	Project management methodology framework documentation
2018-06-23	Newman Configuration
2018-06-30	Coding standards documentation for each repository on the wiki
2018-07-07	Configuration management for local dev and shared environments
2018-07-21	Use Cases and Personas documentation
2018-07-28	Improve test coverage for all repositories to 60%, including AsUnit test suite for sage-scratch
2018-07-28	Complete integration of Intelligent Hinting 1.1 work into SAGE UI
2018-08-04	Chatbot implementation
2018-08-04	Reference Architecture completed
2018-08-11	Intelligent Hinting analysis framework (systematically identify the most effective hint generation and presentation strategies)
2018-08-11	Data visualizations completed

Future Work

Gameful Intelligent Tutoring

- **Context-aware hinting mode selection:** Based on the path progression and wall time of a student in a given puzzle or constructionist game, switch the hinting system between "just-in-time" and "on-demand" hinting. Possibly, decrease the specificity of the hints based on inputs from the outer loop, i.e. how many assignments a student has completed weighted by the "dominant" mastery of each of those assignments.

SAGE Integration

- **Code Review:** Require a code review from another researcher, who has expertise in a relevant technology stack, before merging to the "develop" branch. This would increase collective ownership, help reinforce adherence to the documented coding standards, and improve the overall quality of the code base.

- **Automated team performance metrics tracking:** Automated tracking of team velocity, schedule performance index (SPI), cost performance index (CPI) based on time spent per task, improvement index (weighted average rate of change across velocity, SPI, CPI and other performance metrics). This information is provided back to the team to identify areas for improvement.

Survey, Field Study Design, and Publication Strategy

- **Ongoing Field Studies:** With a framework in place to analyze the efficacy of the hinting system, ongoing field studies would not only provide valuable feasibility and efficacy data based on learning outcomes for students, but could also provide hinting efficacy data that directs future development of the intelligent hinting system.
- **Future publication, case study:** A case-study on SAGE's field studies at various schools would augment the feasibility study, and provide insight into a real implementation of the reference architecture presented in the feasibility study.

References

1) Breiman, Leo. "Machine Learning." *Machine Learning*, vol. 45, no. 1, 2001, pp. 5–32., doi:10.1023/a:1010933404324.

2) Koedinger, Kenneth R., et al. "New Potentials for Data-Driven Intelligent Tutoring System Development and Optimization." *AI Magazine*, vol. 34, no. 3, 2013, p. 27., doi:10.1609/aimag.v34i3.2484.

3) Lwakatare, Lucy Ellen, et al. "Dimensions of DevOps." *Lecture Notes in Business Information Processing Agile Processes in Software Engineering and Extreme Programming*, 2015, pp. 212–217., doi:10.1007/978-3-319-18612-2_19.

4) Mcnamara, Danielle S., et al. "Intelligent Tutoring and Games (ITaG)." *Advances in Game-Based Learning Gaming for Classroom-Based Learning*, pp. 44–65., doi:10.4018/978-1-61520-713-8.ch003.

5) Piech, Chris, et al. "Modeling How Students Learn to Program." *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education - SIGCSE '12*, 2012, doi:10.1145/2157136.2157182.

6) Zhou, Y., Freedman, R., Glass, M., Michael, J. A., Rovick, A. A., & Evens, M. W. (1999, July). Delivering Hints in a Dialogue-Based Intelligent Tutoring System. In AAAI/IAAI (pp. 128-134).

7) Pmi.org. (2018). Agile Approaches on Large Projects in Large Organizations. [online] Available at:

https://www.pmi.org/learning/academic-research/agile-approaches-on-large-projects-in-large-organizations [Accessed 3 Feb. 2018].

8) Brotherton, S. A., Fried, R. T., & Norman, E. S. (2008). Applying the work breakdown structure to the project management lifecycle. Paper presented at PMI® Global Congress 2008—North America, Denver, CO. Newtown Square, PA: Project Management Institute.

9) Feitelson, D.G., Frachtenberg, E., Beck, K.L.: Development and Deployment at Facebook. IEEE Internet Computing 17, 8–17 (2013)

10) Randolph, Gary. "Use-cases and personas: a case study in light-weight user interaction design for small development projects." Informing Science: International Journal of an Emerging Transdiscipline 7 (2004): 105-116 (2004).

11) Maruping, Likoebe M., Xiaojun Zhang, and Viswanath Venkatesh. "Role of collective ownership and coding standards in coordinating expertise in software project teams." European Journal of Information Systems 18.4 (2009): 355-371.

12) Tufte, Edward, and P. Graves-Morris. "The visual display of quantitative information.; 1983." (2014).

13) Razzaq, Leena, and Neil T. Heffernan. "Hints: is it better to give or wait to be asked?." International Conference on Intelligent Tutoring Systems. Springer, Berlin, Heidelberg, 2010.

14) Brewer, Eric A. "Towards robust distributed systems." PODC. Vol. 7. 2000.

15) Cloutier, Robert, et al. "The concept of reference architectures." Systems Engineering 13.1 (2010): 14-27.

16) Angelov, Samuil, Paul Grefen, and Danny Greefhorst. "A framework for analysis and design of software reference architectures." Information and Software Technology 54.4 (2012): 417-431.

17) Piech, Chris and Sahami, Mehran and Huang, Jonathan and Guibas, Leonidas. “Autonomously Generating Hints by Inferring Problem Solving Policies”, Proceedings of the Second (2015) ACM Conference on Learning @ Scale, pp.195-204.

18) Yi Ding, Weimeng Luo, Junyu Zhang (2018). “Intelligent Hinting 1.1 Final Report.”

19) Sutton, R. S., & Barto, A. G. (1998). Reinforcement learning: An introduction (Vol. 1, No. 1). Cambridge: MIT press.

20) Alex Dziena, Johan Sulaiman (2018). “SAGE Field Study Support Final Report”

21) Holton, J. A. (2004). Building trust and collaboration in a virtual team. National Library of Canada = Bibliothèque nationale du Canada.