Eyrie Stakeholder Survey

Eyrie Research Incubator

Hello! The Eyrie Research Incubator is intended to provide a repository of software-intensive projects to support Software Engineering research for Cyber-Physical and Safety-Critical systems -- from inception to maturity.

We are planning a research proposal to submit to the US National Science Foundation and would appreciate your input. Completing this survey should take **less than 10 minutes** of your time.

If you have any questions or concerns, please contact:
Jane Cleland-Huang, University of Notre Dame (JaneClelandHuang@nd.edu)
Robyn Lutz, Iowa State University (rlutz@iastate.edu)

And thank you in advance for your help -- and potential collaboration.

What is the Eyrie Project?

Have you ever been hampered by the lack of publicly accessible software artifacts or runtime experimental environments to support your Software Engineering research? If so -- please help us develop this community resource by answering a few questions about your potential interest in the Eyrie Project.

Eyrie is designed as a research incubator -- to provide a rich repository of Cyber-Physical systems projects in support of core Software Engineering research. Our plan is to initially seed it with two projects -- Dronology , which is a system for managing and coordinating Unmanned Aerial Vehicles (UAVs) and SafeWalk, a system for simulating controls of an astronaut mobility pack. Our plan is to steadily grow the repository with other contributed projects and for Eyrie to provide the infrastructure for the repository's sustainable growth and use.

How does it differ from other Community Resources?

In contrast to existing datasets (e.g., SIRs, PROMISE, CoEST), Eyrie will be populated with rich sets of relatively complete project artifacts (e.g., requirements, design, code, test, safety assurance cases, environmental hazards) across multiple versions of each system, organized as a product line. It will provide rich, shareable, sets of artifacts for experimentation that are typical in Cyber-Physical Systems. Furthermore, all projects within Eyrie will include runtime . environments that are easy to install, reuse, and modify for experimentation purposes

Names and emails have been removed from this report.

Eyrie Stakeholder Responses

Q1: Which of the following best describes your place of employment?			
#	Answer	%	Count
1	Academic	78.79%	26
2	Government	0.00%	0
3	Industry	18.18%	6
4	Other	3.03%	1
	Total	100%	33

Q2 Which region of the world are you from?			
#	Answer	%	Count
1	USA	57.58%	19
2	North America (excluding USA)	9.09%	3
3	Central or South America	0.00%	0
4	Europe	33.33%	11
5	Africa	0.00%	0
6	Asia	0.00%	0
7	Australia or New Zealand	0.00%	0
	Total	100%	33

Q3 - Which of the following areas of research are you actively engaged in?			
#	Answer	%	Count
1	Requirements Engineering	15.19%	24
2	Design	8.86%	14
3	Software Architecture	6.96%	11
4	Testing	7.59%	12
5	Traceability	10.76%	17
6	Formal Methods	8.86%	14
7	Cyber-Physical Systems	10.76%	17
8	Human Computer Interaction	4.43%	7
9	Runtime Adaptation and/or Monitoring	8.23%	13
10	Safety Assurance	10.13%	16
11	Mining Software Repositories	4.43%	7
12	Other	3.80%	6
	Total	100%	158

Other
Cloud Computing
Exploratory Programming (data science based requirements elicitation)
Virtual Agents for Software Engineering
Product Line Engineering
AI and Machine Learning as enablers of more autonomous systems

Q4 - The Eyrie research incubator is designed to support research based on analyzing project artifacts (e.g., traceability, program comprehension) and utilizing runtime environments (e.g., for experimenting with runtime adaptation). What aspects of the Eyrie repository would you be interested in using?			
#	Answer	%	Count
1	Analyzing project artifacts	18.18%	6
2	Using the runtime environment for experimentation	6.06%	2
4	Both aspects (artifacts + runtime)	75.76%	25
5	Neither	0.00%	0
	Total	100%	33

Q5 - What are your primary sources of data today? (Please check all that apply)			
#	Answer	%	Count
1	Open Source Systems	21.43%	21
2	Existing community repositories (e.g., SIRS, PROMISE, COEST)	9.18%	9
3	Industrial data sets from collaborators	19.39%	19
4	"Homegrown" datasets	21.43%	21
5	Examples taken from published case studies	11.22%	11
7	Data sets released in conjunction with a published paper, but not part of a formal collection	14.29%	14
6	Other	3.06%	3
	Total	100%	98

Other
My own industrial datasets
Using city transportation data for smart city applications
Actual product data

Q6 - To what extent have you used the following types of artifacts in your research over the past 3 years:								
#	Question	Never		Sometimes		Often		Total
1	Requirements	6.45%	2	29.03%	9	64.52%	20	31
2	Software Architecture	35.48%	11	45.16%	14	19.35%	6	31
3	Design Specifications	19.35%	6	35.48%	11	45.16%	14	31
4	Design Rationales	51.61%	16	32.26%	10	16.13%	5	31
5	Test Cases	29.03%	9	25.81%	8	45.16%	14	31
6	Source Code	16.13%	5	35.48%	11	48.39%	15	31
7	Safety Artifacts (e.g., Safety Assurance Cases, FMECA)	38.71%	12	38.71%	12	22.58%	7	31
8	Formal Specifications	51.61%	16	22.58%	7	25.81%	8	31
9	Models	25.81%	8	22.58%	7	51.61%	16	31
10	Trace Links	29.03%	9	29.03%	9	41.94%	13	31
11	Feature Models	54.84%	17	25.81%	8	19.35%	6	31
12	Other	87.10%	27	0.00%	0	12.90%	4	31

Q7 - How satisfied are you with the current availability of project data to support your research?			
#	Answer	%	Count
1	Extremely satisfied	0.00%	0
2	Somewhat satisfied	38.71%	12
4	Not satisfied	61.29%	19
	Total	100%	31

Q8 - Do you agree with the following statement: "My research agenda has been influenced by availability of appropriate data sets"			
#	Answer	%	Count
1	Strongly agree	64.52%	20
2	Somewhat agree	35.48%	11
3	Disagree	0.00%	0
	Total	100%	31

Q9 - Which of the following issues have hindered your research with respect to data sets? (Leave blank if you have not experienced any problems)			
#	Answer	%	Count
2	Missing or incomplete artifacts (e.g., informal issues were served as proxies for requirements)	22.69%	27
3	Poor quality data (e.g., trace links not validated)	17.65%	21
4	Data set is a snapshot taken from one version (e.g., requirements from version x) and that is a problem because my research requires multiple versions.	12.61%	15
6	Data set is not industrial quality (e.g., created purely by students)	23.53%	28
8	Data set is trivially sized	19.33%	23
7	Other:	4.20%	5
	Total	100%	119

Other

Datasets are focused on a particular aspect of software development (e.g., RE), no end-to-end coverage of entire development lifecycle
Confidentiality
Smart city data is often held as proprietary by companies; ot even city can access it
Product data often is not well organized and difficult to abstract
Datasets not properly cleansed of noise e.g. software clones

Q10 - To what extent are you currently (or plan in the near future) to engage in projects that provide APIs to support:								
#	Question	No engagement		Planned Engagement		Current Engagement		Total
1	Self-Adaptation	48.15%	13	33.33%	9	18.52%	5	27
2	Runtime monitoring	37.04%	10	37.04%	10	25.93%	7	27
3	Task/Agent Coordination	62.96%	17	18.52%	5	18.52%	5	27
4	AI decision making	29.63%	8	37.04%	10	33.33%	9	27
5	Other	92.59%	25	0.00%	0	7.41%	2	27

Q11 - Which of the following techniques do you use to currently evaluate research with runtime needs? (Check all that apply)			
#	Answer	%	Count
1	Worked examples	26.09%	12
2	Prototype implementations as proofs-of-concept	36.96%	17
3	We use our own research system (please briefly describe):	13.04%	6
4	We use a publicly available system (please name it):	6.52%	3
5	We use an industrial system provided by collaborators	13.04%	6
6	Other	4.35%	2
	Total	100%	46

Other

We use our own research system (please briefly describe): - Text
Geocast Air Operations Framework - system and platform for safe and secure UAS operations, communications, and coordination at large scale
Eclipse Capra
student projects
Rainbow, AcmeStudio
ReMinds Requirements Monitoring Framework
Reactive Mission Planner for Fixed Wing UAVs in the domain of agriculture data collection

We use a publicly available system (please name it):

Turtlebot/ROS
An urban traffic simulator called CrowdNav: https://github.com/Starofall/CrowdNav
AFRL's Open UxAS

Q12 - Are existing runtime environments sufficient to support replicable, reproducible, generalizable research?			
#	Answer	%	Count
1	Yes	4.00%	1
2	Partially	44.00%	11
3	No	52.00%	13
	Total	100%	25

Q13 - What problems have you experienced in finding, using, and sharing, runtime environments for research purposes.
I'm not aware of any runtime requirements-to-code environment that would allow us for real-scale experimentation related with our work on obstacle-driven runtime system adaptation and our work on runtime adaptation in view of environment changes.
Usability and compatibility. Lack of documentation/dependencies.
Most of them are not user-friendly and the documentation often available is not sufficient.
Because finding artefacts for cyber-physical systems currently requires industrial collaborations, we are not allowed to share them with anybody, including academic collaborators. Having a publicly available benchmark would help.
Installing other peoples systems often is very difficult due to specific constraints
It is very costly to either develop our own, or figure out how to reuse others'. Design rationale and architecture is usually missing.
We are still in a period of proprietary systems. For CAV, each automaker wants to build their own system that only works for their own cars. Transportation infrastructure systems (signal controllers, parking monitors) have same problem: companies hold their data close to the vest. This includes companies like Google Waymo. I think it will eventually change but for present it is a problem. Another issue in our smart city work is human-in-the-loop simulation. Even if we have data, we lack open simulators that can use the data to run realistic simulations. My work is looking at sustainable transportation, AKA active transportation. We are finding we have to build our own simulation lab to do research in this area.
Poor documentation of the APIs Lack of trust in the implementation because of poor coverage with associated tests
all the same ones you've mentioned on the previous page
Sorry - not sure what you mean by "runtime environment"
Intellectual Property Rights make sharing of tools and data difficult
Honestly, I don't even know where to start looking. We primarily use our own prototypes.
Lack of clear interfaces Lots of customization was required to adapt the prototype The runtime simulated environment is not recognized as a credible surrogate for a real case study
Docker has alleviated some of these issues, but not everyone provides docker containers.
Unavailability of simple to set up environments.
Knowing what is available. When something is available it often isn't well documented.

Q14 - Which of the following programming languages would be acceptable for your own experimentation?								
#	Question	Unacceptable		Acceptable		Preferred		Total
1	C	7.41%	2	81.48%	22	11.11%	3	27
2	C++	11.11%	3	81.48%	22	7.41%	2	27
3	C#	7.41%	2	81.48%	22	11.11%	3	27
4	Python	0.00%	0	59.26%	16	40.74%	11	27
5	Java	0.00%	0	55.56%	15	44.44%	12	27
6	Other	0.00%	0	20.00%	1	80.00%	4	5

Q15 - Do you envision yourself as a user of the Eyrie repository?			
#	Answer	%	Count
1	Probably Yes	96.55%	28
3	Probably No	3.45%	1
	Total	100%	29

Q16 - Do you envision yourself as a potential contributor to the Eyrie repository? If so, please check all potential contributions.			
#	Answer	%	Count
1	Contributing towards improving or enriching existing sets of artifacts associated with Dronology, Safewalk, or future Eyrie Projects.	18.18%	14
2	Contributing a new or existing project to the Eyrie repository	20.78%	16
3	Serving in an advisory capacity on the industry advisory board	11.69%	9
4	Serving on a research advisory board	24.68%	19
5	Actively contributing towards governance of the Eyrie Community (e.g., reviewing new project submissions, providing feedback, alpha/beta testing, helping to maintain references to papers that utilize Eyrie projects.	9.09%	7
6	Contributing to a core Eyrie project (i.e., committing code back to the project that you developed for your own experiment)	15.58%	12
	Total	100%	77

Q17 - Could we include your name in our NSF proposal as part of a list of potential Eyrie stakeholders?			
#	Answer	%	Count
172	Yes	100.00%	32
173	No	0.00%	0
	Total	100%	32

Q18 - As a final question, do you have any general suggestions for us as we develop the Eyrie research incubator? What would make it useful for you?
As a final question, do you have any general suggestions for us as we develop the Eyrie research incubator? What would make it useful for you?
A key challenge is representativeness and trustworthiness of the datasets. Is a case study included in Eyrie representative of real systems? There are many covariants here, people that develop the system, development environment, languages, programming technologies and etc. In my research, I often need more than one representative datasets but are build by different minds and in different development environments and constraints. Otherwise, I don't observe the same defects and root causes of the defects that I'm looking for in these case studies (in the context of one of my research projects) Second trustworthiness, Open source projects actually work very well for me, many of the projects that we use the source code are public but these projects are built by companies internally and for certain business motivations they have been released to the public. Any dataset used in Eyrie should be assessed in terms of trustworthiness, who developed it, and in what context? Is it yet another artificial case study (or project from academia) or has all the criteria of a robust and reliable project?
In our research on requirements-driven runtime system adaptation, we would use the software artifacts, source code, and executable environments that Eyrie intends to develop and make available in order to experimentally evaluate our own research in goal-oriented model construction, analysis, implementation, and adaptation (at design time and at runtime). Our research community desperately needs real, well-documented systems, as opposed to toy examples, as common benchmarks for evaluating and comparing research results.
Maybe include quantitative metrics / benchmarks. It's one thing to have data, it's another to define objectives to reach
If the repository for projects is to contain source code, providing the (optional) capability to publish a live distribution (i.e. one where ongoing development and contributions can be actively made and supported) would be excellent; i.e. basically a git repository that can be effectively an open source project base. That is as opposed to a simple static archive of a single project snapshot.
It would be useful to make sure all artifacts are well documented.
Make it accessible (like in accessibility) to graduate students.
It would be great to build a strong community surrounding for Eyrie search projects towards common research goals. It's also important to maintain the projects held by Eyrie up-to-date.
A configurator for Eyrie would be extremely helpful. A student starting to work on a project could select the datasets they need along with the required tools and be able to download a virtual machine that they can easily get running. In general, a VM concept would be preferrable. That would also allow sharing the final version of the artifacts and tools and the evaluation environment with other researchers and include a link in papers. The ML community requires a fully functional and documented VM for most submissions already. If Eyrie would support the use of VMs out-of-the-box, it would be much easier to establish a similar standard in the RE/SE community.

I need the development history like issue tracker work items, ideally with links to code. Requirements in an issue tracker (or as user stories) instead of long requirements documents
Good documentation. Available support staff to answer questions, assist with installation, etc.
I think Eyrie sounds very useful; however as my circumstances have changed, it is no longer in line with the work that I am currently doing, making me a less-likely contributor/user (though no less enthusiastic about the direction).
I've got the impression that it already has the potential to be useful.
I have two projects. One is using formal methods to look at smart city applications and cybersecurity holes. For this project, I could use the type of content that I think Eyrie can provide. The other project is on supporting urban planners and city traffic engineers in terms of smart transportation and CAV applications. At moment this falls under what I would call exploratory programming. It's data science focused. What it requires is diverse data sets and the ability to run simulations to complement field evaluations. I don't think Eyrie is a fit for this. But could be wrong :)
Document! APIs, design rationale, what it is/isn't supposed to do. Include defect reports and link them to commit information.
you guys are doing a great job. best wishes
I am a little fuzzy on your vision, but look forward to seeing what you are up to. Most of our research is funded by the government, and we generally release open-source models/tools, for example on github, so that other research groups are able to replicate our stuff. In some cases (such as on large DARPA programs) the program's research performers share tools and models with us and we do likewise in order to leverage each others work. It would be interesting to understand how your approach would improve upon that kind of tactical collaboration.
Maintenance becomes challenging if many research groups use the research incubator in a clone-and-own manner. It is important to set up a communication channel ensuring outside contributions while keeping the platform consistent.
I strongly support your ambitious plans. Two things come to my mind. First, don't forget about the value Eyrie could have for education. Software and systems engineering courses around the world could potentially make good use of the content. Second, it might be a good idea to reach out to researchers with a strong background in critical systems - I suggest the "Critical Systems" researchers at Univeristy of York, UK. https://www.cs.york.ac.uk/research/research-themes/critical-systems/
Would be great to expand to the automotive domain and cover both autonomous and connected vehicles. Also interesting to consider the appropriate safety standards in the project and include them in the artifact repository.
It would be great to create a synergy with other initiatives, e.g., in Europe and in Asia. While the proposals will be submitted independently, the establishment of a worldwide network would significantly increase the chances for the incubator to gain widespread adoption
Need to understand it a bit better. It seems promising.
As with most things, key to adoption is to keep the ramp up as short as possible.
Examples of use are always helpful. A couple of the questions mentioned education - I'm always interested in finding ways to expose students to larger code bases. I could see getting students working with code as part of a class and then transitioning them into a research project.